AD-A 034 427

36

TECHNICAL
LIBRARY

RECALL:

A Management Information Retrieval System

for the Wang 2200

November 1976

U.S. Army Materiel Development
and Readiness Command

HARRY DIAMOND LABORATORIES

Adelphi, Maryland 20783

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER HDL-TM-76-36	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* RECALL: A Management Information Retrieval System for the Wang 2200		S. TYPE OF REPORT & PERIOO COVEREO Technical Memorandum
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Howard M. Bloom		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND AODRESS Harry Diamond Laboratories 2800 Powder Mill Road Adelphi, MD 20783		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME ANO AODRESS Commander US Army Armament Command Rock Island, IL 61201		12. REPORT OATE <u>November 1976</u> 13. NUMBER OF PAGES 61
14. MONITORING AGENCY NAME & AOORESS*(if different from Controlling Office)*		1S. SECURITY CLASS. *(of this report)* Unclassified
		15a. OECLASSIFICATION/OOWNGRAOING SCHEOULE
16. OISTRIBUTION STATEMENT *(of this Report)* Approved for public release; distribution unlimited.		
17. OISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*		
18. SUPPLEMENTARY NOTES HDL Project: 738671 DRCMS Code: 738017.H0.0R000		
19. KEY WOROS *(Continue on reverse eide if necessery and identify by block number)* RECALL Wang 2200 Data-base system Programmable desk calculator	Storage and retrieval	

20. ABSTRACT *(Continue on reverse side if neceseary end identify by block number)*

A data-base language called RECALL has been implemented on the Wang 2200 programmable desk calculator. The language was patterned after RETRIEVE, developed by Tymshare, Inc., for its time-sharing network. Differences between the two implementations are very minor. The report describes each command in the language and gives a comprehensive example to illustrate how the entire system can be used. The listing of the program allows the reader to make modifications, if necessary. The RECALL system is

DD FORM 1 JAN 73 **1473** EDITION OF 1 NOV 65 IS OBSOLETE

implemented on a 2200C calculator with 16k bytes of memory and a model 2230 disk used for temporary storage of the overlay segments needed for runing the program. The system assumes that the data bases will be permanently saved on tape cassette.

CONTENTS

APPENDICES

FIGURES

TABLES

1. INTRODUCTION

A data-base language (DBL) called RECALL has been implemented on the Wang 2200 programmable desk calculator. The language was patterned after RETRIEVE, originally designed for the Tymshare computer system. Because of the ease in using the language, it was believed that RETRIEVE would be an ideal DBL to use with the Wang system. Hence, the language has been implemented in its entirety with just a few small changes. This report only briefly describes the language itself. Most of the emphasis is placed on how to run the system on the Wang 2200 and what the differences are between the original Tymshare version and the new Wang version. The RETRIEVE[1] manual details the language.

The Wang RECALL system has been implemented on a 2200C calculator with 16k bytes of memory. The system uses temporary storage on the last 320 sectors of both the fixed and the removable disk platters on the model 2230 disk. The system's main program requires approximately 12k bytes for the common subroutines and variables. Four 3k-byte segments are overlaid when necessary for the implementation of the various commands. The various segments are stored on the fixed disk platter. The system assumes that the data are saved on tape cassette and is modelled around this concept.

A technique for using RECALL data bases in the BASIC program is described in appendix A. The source code for RECALL is listed in appendix B.

2. DATA-BASE DESCRIPTION

The RECALL system is designed to store a given collection of related data and to access and update this information, the "data base." A data base is divided into "records"; each record consists of one or more "fields"; each field contains a datum such as a number or a string of characters. All records in a data base contain the same number of fields arranged in the same order.

A field may contain data from 1 to 32 characters in length, and a record may contain up to 120 characters. There can be a maximum of 15 fields in a record. The entire data base can contain approximately 78,000 characters, i.e., the amount of storage on a Wang 1251 tape cassette. The data-base name can be one to eight characters in length.

---

[1]*RETRIEVE, Tymshare, Inc. Palo Alto, CA (1971).*

A field is referenced by its name, which can be one to eight characters. It must begin with a letter and can contain only letters, digits, and periods. The following names cannot be used because they are reserved for commands: AND, FOR, FROM, IN, NOT, ON, OR, and WITH.

A field can be of either character or numeric type. A character field can contain any character on the keyboard, whereas a numeric field can contain only numbers. A character field is specified by the letter C when defining the data base. There are two kinds of numeric fields, integer and noninteger. An integer field cannot contain decimal points, whereas a noninteger field can. However, a noninteger field cannot use the E format number (allowable in Tymshare). An integer field is specified by the letter I; a noninteger field, N.

2.1 Command Description

The RECALL system is operated by the specifying of a set of commands (tables I to IV) to manipulate the data base. The general format of a command is the following:

[range list]   command   [FOR condition]

Only the first three characters need be typed for all commands except REPORT (REPO must be used). The range list allows the user to specify which records within the data base should be considered for that command. The list can be of three types:

a. A single record number (e.g., "45")

b. A range of records, consisting of a pair of numbers separated by a colon (e.g., "15:20" means records 15 to 20)

c. Any combination of single number or range; the record numbers are separated by semicolons (e.g., "15;18:20;25" means records 15, 18, 19, 20, 25)

Up to five combinations (four semicolons) can be specified. If a range list is not specified for a given command, then it is assumed that the whole data base is to be used.

The Wang implementation of RECALL uses a semicolon rather than a comma as a separator. Warning: A comma should never be typed for any command within the RECALL system. The blank space is not acceptable as a separator for the range list. The "$" cannot be used to specify the last record.

TABLE I. DATA BASE CREATION AND ACCESS COMMANDS

Command	Function
CREATE data base	Creates new data base with given name (destroys presently active data base in system)
APPENO (FROM data base)	Allows records to be added to data base from keyboard or from data base on tape
LOAO data base	Loads data base from tape into storage as active data base
MERGE ON field list (FROM data base)	Merges data base on tape with present data base
[range] SAVE TO data base [FOR option]	Saves present data base on tape
STRUCTURE	Oisplays field structure of data-base records
SIZE	Oisplays number of records in data base
QUIT	Ends RECALL program

TABLE II. DISPLAY COMMANDS

Command	Function
[range] LIST [field list] [FOR option]	Oisplays data-base records for specified fields with record numbers and field headings included
[range] PRINT [field list] [FOR option]	Same as LIST, except no record numbers
[range] FAST [field list] [FOR option]	Same as PRINT, except no heading
[range] COUNT [FOR option]	Counts number of records
[range] SUM expression list [FOR option]	Oisplays data base totals for each expression (five max)
[range] AVERAGE expression list [FOR option]	Displays data-base average values for each expression (five max)

TABLE III. UPDATING COMMANDS

Command	Function
[range] CHANGE [field list] [FOR option]	Allows selected fields to be changed in record
[range] DELETE [FOR option]	Deletes selected records
[range] REPLACE field$_1$ with expression$_1$ [;field$_2$ WITH expression$_2$;. . .] [FOR option]	Replaces selected fields with any desired expression
SORT ON field list	Ascending sort of data base

TABLE IV. REPORT GENERATION COMMANDS--[RANGE LIST] REPORT ["FOR" CONDITION]

	RECALL prompt	User responses allowed
1	REPORT OUTPUT TO	T (to display) or data base name (to save on tape)
2	REPORT FORM NAME	Name (if blank, skip to 4)
3	UPDATE REPORT FORM	Y or N (if N, old report form is used)
4	HEADING	Y or N
5	DOUBLE SPACE	Y or N
6	TOTALS	Y or N (if N, skip to 10)
7	SUBTOTALS	Y or N (if N, skip to 9)
8	BY ITEMS	List of items (record fields) on which to subtotal (three fields max)
9	SUMMARY REPORT ONLY	Y or N
10	COL WIDTH:CONTENTS 1 . . .	Column width; expression (requests description of each column in report--end with RETURN to column prompt (15 columns max)
11	COL HEADING 1 . . .	Heading (requests column headings) (10 characters per row max, three rows max)
12	COLUMNS-TOTAL; NO OF DECIMAL PLACES column number prompt	Y or N; number of decimal places (prompted for each numeric column) (five columns max)
13	COLUMNS NO OF DECIMAL PLACES column number prompt	Number of decimal places (asked if totals were not requested)

2.2 Expressions and Conditions

The FOR condition allows the user to further specify on which records a command is to operate. For example, LIST FOR SIZE > 100 lists all records in the data base for which the field name SIZE has a value greater than 100.

A condition is a combination of expressions. Hence, first expression must be defined.

A numeric expression can be any of the following:

a. A number

b. A numeric field name

c. Any meaningful combination of number and numeric field name, using arithmetic operators

The following operators are allowed: ↑, *, /, +, and -. Warning: Unary negation is not allowed. One must use "0-5" to represent "-5."

A character expression can be any of the following:

a. A string enclosed in single quotation marks (Double quotation marks are not allowed.)

b. Any character field name

c. Any meaningful combination of string and field name, with the operator + used for string concatenation

A condition may consist of numeric expressions or character expressions related by these relational operators: < (less than), > (greater than), = (equal to), # (unequal to), <= (less than or equal to), >= (greater than or equal to).

The long form (e.g., LESS THAN for <) allowable in Tymshare RETRIEVE was not implemented in Wang RECALL.

Another condition consists of character expressions related by the operator IN (or NOT IN). This operator specifies that one string is (or is not) contained in another.

The relational expressions can be combined by using the following logical words, listed in descending order of precedence:

9

Word	Meaning
NOT	NOT A is true if A is false.
AND	A AND B is true only if A and B are both true
OR	A OR B is true if either A or B or both are true.

Conditions in RECALL are always preceded by the FOR modifier and always have a value of true or false.

Parentheses may be used in expressions and conditions to specify the order of operations. For example, "(4+3)*5" has the value 35, but "4+3*5" has the value 19.

In evaluating an expression or condition, three rules govern the order in which operations are performed:

a. All operations with the innermost set of parentheses are performed first; then those within each succeeding outer set are performed.

b. Without violating rule a, operations are performed in the following order:

$\uparrow$

* and /

\+ and -

<, <=, >, >=, =, #

IN, NOT IN

NOT

AND

OR

c. Without violating rules a and b, operations are performed left to right.

2.3 Line Continuation

If there is not enough room to enter the entire command (or input data record) on one line, the user can type the character "&" and then proceed to the next line. If this character is not used and the command runs over into the next line, the remaining portion of the command is ignored. A field, name, number, or string cannot be continued from one line to the next. The user should backspace to erase the information just entered, then type the "&," and begin the information on the next line.

2.4 Field and Expression Lists

The field list contains one or more field names separated by semicolons. (In Tymshare, commas are used.) The expression list contains one or more expressions separated by semicolons.

2.5 Running RECALL System

The user should check that the equipment is turned on, including the master switch for the central processing unit (CPU) and cathode ray tube (CRT). Someone authorized must turn on the disk. If he desires hard copy, the user should turn on the line printer (depress the POWER and SELECT buttons).

The system is loaded by keying

    LOAD DCF "RECALL"

    RUN

To end every command, the RETURN button is keyed.

The system responds with this message:

    RECALL SYSTEM-REVISION 8/22/75

    DATE (MM/DD/YY)?

The first line gives the present revision date of the system, and the second line requests the present date. This date subsequently is used if a report is generated or a data base is saved on tape. The user then types in the date, using the digit code for the month, day, and year.

The system is now ready for the user to manipulate the data base by typing commands. The system responds "COMMAND?" and the user inputs the desired command. The system performs the appropriate operations and then prompts for a new command. If the format of the command is wrong, the system prints "ILLEGAL COMMAND" and the user must reenter his command. When the user is finished with the system, he should enter the command QUIT. This command halts the system.

2.6 Hard-Copy Capability

The system is designed to generate all displays at the CRT. If the user wishes to print out on a hard-copy unit, he should respond to the command prompt by keying HALT/STEP and then typing in the desired command. The system responds with a colon. The user can type

SELECT PRINT XXX(YYY)

to indicate the hard-copy unit XXX (e.g., "215") and the column width YYY (e.g., "132"). He then keys "CONTINUE." The system generates the printout at the hard-copy unit. When he wants to return the display to the CRT, the user can again respond to the command prompt by keying HALT/STEP and then the command. After that, he should type this:

SELECT PRINT 005(64)

2.7 Data-Base Maintenance

The system always operates on the active data base that is stored in the memory and on a disk. Whenever any operation is performed to alter the data base (e.g., sorting or deleting), the active data base is changed and the previous state is destroyed. Warning: The user should save pertinent copies of the data base on tape before commanding a status change. In Tymshare, the old version of the data base is saved under a different name, and the user is asked if he wishes to maintain this old version.

2.8 Tape Handling

For all commands requiring tapes, the system prints the message:

MOUNT TAPE AND TYPE CONTINUE

2.9 Differences between Tymshare RETRIEVE and Wang RECALL

The Tymshare (T) RETRIEVE and Wang (W) RECALL differ as follows:

12

a. T uses a comma or blank as a field separator; W, a semicolon.

b. T uses $ for the last record indicator; W does not.

c. In T, names have 10 characters; fields, 24; and records, 185; there can be 18 fields. In W, names can be only 8 characters; fields 32; and records, 132. There can be only 15 fields.

d. A floating point numerical format (E) is allowed in T, but not in W.

e. The ALL, BINARY, IF, RECNO, RESULTS, SCRAMBLED, SEQUENCED, FIELDS, and SYMBOLIC key words have been implemented in T, but not in W.

f. The commands SORT and MERGE use BY in T, but ON in W.

g. The command BASE is used in T, but only LOAD is used in W.

h. SORT and MERGE lists need not be specified and can be greater than three fields in T; SORT and MERGE lists must be specified and cannot be greater than three fields in W.

i. The operator NOT can be used with IN and OUT in T, but only with IN in W.

j. The long form for relationships in T is not used in W.

k. Lines are continued by using the line continuation key in T, but the operator "&" in W.

l. In T, 10 columns max can be totalled in the command REPORT, and 10 fields max can be used to determine subtotals. In W, five columns max can be totalled in the command REPORT, and three fields max can be used to determine subtotals.

m. In T, 10 fields max and, in W, 5 fields max can be summed or averaged by using the command SUM or AVERAGE.

n. In FOR conditions, string constants can use double quotation marks in T, but only single quotation marks in W.

o. Editing is done with control characters in T, but with the BACKSPACE and LINE/ERASE keys in W.

p. An old version of a data base saved in T is not saved in W when a command such as SORT is executed.

q. The command LOAD can be used to create a new data base in T, but not in W.

r. The space between fields generated in the command REPORT when saving the report in a file is included in the field width in T, but not in W.

s. The expression list used in the commands SUM and AVERAGE need not be specified in T, but must be in W.

t. The command MODIFY is implemented in T. The command CHANGE must be used in W.

u. In T, the old value is retained when CHANGE is commanded. In W, each field specified for an edit in the command CHANGE must be specifically entered when prompted; if not, a blank field is stored.

v. The headings on a report can have 20 characters max for any column on any line in T, but 10 in W.

3. DATA-BASE CREATION AND ACCESS COMMANDS

Certain commands create and access the data base.

3.1 Creating Data Base

The command CREATE [data base] creates the data base. If the user does not specify the data-base name (one to eight characters), the system prompts with "DATA BASE?" RECALL does not allow either binary or scrambled data bases as RETRIEVE does.

As soon as the command is entered, any data base active in the system is destroyed. The system prompts with

PLEASE TYPE IN THE STRUCTURE OF YOUR DATA BASE

FIELD   NAME;WIDTH;TYPE;DECPL

The system then prompts with a field number, and the user enters the desired name; width; type (I, N, or C); and, if N, the number of decimal places. If the user prematurely ends the line without entering all the information or enters incorrect information, the system prompts for each piece of information separately until the entire field is entered. The user responds to the FIELD prompt by keying the command RETURN when all the fields have been entered.

14

To prompt for the actual record data, the system first displays the sequence of field names. The user enters each record that ends with a command RETURN. He ends the record prompt by keying RETURN at the beginning of a record. (The character / symbolizes RETURN in this report.) If the user prematurely ends the record without entering all the fields or if he enters an incorrect field value, the system prompts for the remaining field values in the record by first displaying the field name. The record can be continued on the next line by using the operator "&." The system prints the number of records stored for the data base, as in this example:

```
1    CUSTOMER;15;C /
2    AGE;3;I /
3    PRICE;8;N;2 /
4       /

CUSTOMER;AGE;PRICE

APPLE;20;18.10 /
PEAR;14;22 /
GRAPE;84;171.45 /
   /
3   RECORDS
```

The values of all character fields are stored left justified in the data base. The values of the numeric fields are stored right justified. If a numeric field is specified as having three decimal places, then, when the value is entered, the appropriate number of trailing zeros is added (with a possible decimal point) before the value is stored in the data base.

The data base is now defined and can be manipulated by all the other commands. However, this active data base does not have any permanent storage. Only the command SAVE allows a data base to be saved.

3.2 Adding to Data Base

The command APPEND (FROM data base) allows the user to add records to the present data base. If the "FROM [data base]" is excluded, the system prompts for the records to be entered from the keyboard as in the command CREATE. If a data-base name is included, the system prompts with this:

MOUNT TAPE AND TYPE CONTINUE

The user loads the tape, and the system reads all the records from the tape and adds them to the active data base. The records must be in the exact format as the active data base. If the name on the tape disagrees with the name in the command, the system prints "WRONG DATA BASE" and indicates an illegal command. After all the records are loaded, the system prints the new record size of the data base. The RETRIEVE modifiers SEQUENCED and FIELDS are not implemented in RECALL.

3.3 Loading Data Base

The command LOAD [data base] is used to load a data base that has previously been saved on tape. If the name on the tape disagrees with the name on the command, the system prints "WRONG DATA BASE" and indicates an illegal command. The system displays "DATABASE [name] LAST SAVED [date] HAS BEEN LOADED."

If the user does not type the data-base name, the system prompts with "DATABASE?"

This command differs slightly from that in RETRIEVE: in RECALL, the command can be used only to load a previously created data base, not to create a new data base (also, the word BASE cannot be used as a substitute for LOAD).

3.4 Merging Data Bases

The command MERGE ON [field list] FROM [data base] allows the user to merge data from a data base stored on tape into the active data base. The two data bases must be presorted in the order that is desirable for the merge. The two data bases also must have identical record structures. The field list can contain from one to three fields in the order of the desired merge, as in this example:

MERGE ON AGE, SALARY FROM PERSON

(In RETRIEVE, the ON field list is optional.) In the example, the data base PERSON is merged with the active base with respect first to AGE and then to SALARY.

3.5 Saving Data Bases

The command SAVE TO [data base] saves the active data base on tape. It is one of the commands that can use only a selected set of records. The system prompts with

MOUNT TAPE TO SAVE DATABASE—TYPE CONTINUE

and then saves the selected set of records.

3.6 Displaying Data-Base Structure

The command STRUCTURE is used to display the field description of the records. The heading

FIELD; TYPE; WIDTH; NAME

is displayed, and then the field information is given. If the field is numeric (N), the width includes the number of decimal places.

3.7 Displaying Data-Base Size

The command SIZE displays the number of records in the data base.

4. DATA-BASE DISPLAY COMMANDS

Commands are available for displaying selected information from the data base. All commands can have a record range list and FOR condition.

4.1 Displaying Set of Records

The commands LIST, PRINT, and FAST are all used to display a set of records. The command LIST prints out the field names as a heading and includes the record numbers. The command PRINT does not have record numbers, and the command FAST does not have record numbers or a heading. If no field list is given, all fields for each record are displayed in the order that they are stored. The field width is used to determine the space used in displaying each field. One extra space is placed between each field for clarity. If a field list is given, the display contains the data values for each record in the order that they are specified in the field list. The number of records displayed is given after the records are displayed.

For example, if the data base contains the fields CUSTOMER, AGE, and PRICE and record 5 contains CITRUS;45;21.23, the command 5FAST produces this:

CITRUS   45   21.23

The command 5FAST PRICE;AGE produces this:

21.23   45

4.2 Counting Records

The command COUNT is used to determine the number of records that satisfies a given condition. The system displays the record count. Without the FOR condition, the command COUNT is identical to the command SIZE. This is an example:

    COUNT FOR SALARY>4500 AND AGE<29

4.3 Summing Expressions

The command SUM expression list is used for totalling specified numeric fields or expressions. There can be five expressions max in the list. For example,

    SUM SALARY;AGE*SALARY FOR AGE>30

computes the sum for all salaries and the sum of the product of the age times the salary for all records where the age is greater than 30.

The system prints out the expression list before displaying the sums. The number of records used also is displayed. Unlike in RETRIEVE, the expression list must be specified and only numeric fields can be used in RECALL.

4.4 Averaging Expressions

The command AVERAGE expression list is used for averaging expressions. The usage is identical to that of the command SUM.

5. DATA-BASE UPDATING COMMANDS

Commands update the information in the data base.

5.1 Changing Record Fields

The command CHANGE [field list] allows the user to change selected records or selected fields within a record. If the field list is not specified, the entire record is updated. If the list is given, only those fields are updated. For the range of specified records, the system displays the following:

    [field name] [old value]?

18

The user enters the new value. This prompt is repeated for each specified field in the record. As an example, assume that the record for the customer APPLE contains 32 as his age, and the value should be updated to 35.

CHANGE AGE FOR CUSTOMER = 'APPLE'

The system responds with the following line up to the colon:

AGE   32? 35

The user responds with "35," and the record is updated. The command MODIFY (the same as the command CHANGE, but without the old value displayed) is not implemented.

5.2 Deleting Records

The command DELETE deletes a selected set of records based on the range list and FOR condition. If both options are omitted, the entire data base is deleted.

5.3 Replacing Records

The command REPLACE [field$_1$ WITH expression$_1$] [;field$_2$ WITH expression$_2$; . . .] allows the user to replace selected fields with any desired expression. The command is especially useful when the user wishes to change a number of records in the same way, since the user need not enter the changes for each record separately as for the command CHANGE. For example,

REPLACE AGE WITH AGE*2

doubles the value of all ages in the data base. AGE is immediately updated, so that if it appears in a later expression in the command REPLACE, the new value is used.

5.4 Sorting Records

The command SORT ON [field list] allows the user to perform an ascending sort on as many as three fields in a data base. The field list must be included. (It may be omitted in RETRIEVE.)

6. REPORT GENERATION

The command [range list] REPORT [FOR condition] initiates report generation. If neither the range list nor FOR condition option is used, all the records in the data base are reported.

After the command is issued, the system dialogues with the user (table IV) after first requesting the user to mount a report tape.

a.   If the user responds T, the report appears on the line printer. Otherwise, the report is saved on tape as a data base with the specified name. All the records specified in the command are saved using the fields defined in prompt (table IV). This tape can then be used to set up a new data base. The user creates the new data base with the command CREATE and then uses the command APPEND to read the records from the tape. This feature is especially helpful in restructuring a data base.

b.   The name specified by the user is shown on all report forms.

c.   If the answer is NO, the system gathers all the necessary information from the report tape.

d.   If the answer is YES, the system contains column headings and is formatted into pages 8-1/2 × 11 in. (21.6 × 27.9 cm).

e.   If the answer is YES, only the body of the report is double spaced; headings and totals are single spaced.

f.   If the answer is YES, the system accumulates totals of report columns containing numeric data. The user specifies which columns of the report are to be totalled by responding to the COLUMNS TOTALS question. Five columns max can be totalled.

g.   If the answer is YES, the report shows subtotals for numeric report columns each time that the value of a selected data-base field changes.

h.   A list of fields (separated by semicolons) is given for which, after a change in value for any field, a subtotal is generated. Three fields max can be specified.

i.   If the answer is YES, only total and subtotal headings and values are shown on the report. Individual records do not appear.

j.   The column contents are specified. The user ends the prompt with the command RETURN following the column number, as in this example:

```
COL

1 WIDTH;CONTENTS  8;LOT/
2 WIDTH;CONTENTS  8;PRICE*QTY/
3 WIDTH;CONTENTS/
```

Fifteen columns max can be specified. The system displays a blank space between each column. Unlike in RETRIEVE, these extra columns do not count when a data-base tape is generated.

k.   The column headings are specified. The system prompts with the column number. By being separated with slashes, the heading can be spread over three lines. Each line can be 10 characters max, as in this example:

```
COL HEADING

1    LOT/IDENT/
2    COST/
```

The actual page heading looks like this:

```
LOT    COST
IDENT
```

1.   The columns' totals and decimal place information are requested for all the numeric fields. The system prompts with the appropriate column number, as in this example:

```
COLUMNS-TOTALS; NO SPACE OF DECIMAL PLACES

2 Y;2/
```

The number of decimal places must be specified. If it is specified as zero, the field is considered integer. In this example, column 1 is not prompted. This field is character and not numeric.

m.   This question is asked only if the user has not requested totals. During the dialogue, if the display output feature is chosen, the program is stopped, and the user is requested to type the select print option desired.

If the heading has been requested, the following block of information appears in the upper left corner of every page.

```
PAGE [number]
DATE: [MM/DD/YY]
DATABASE: [name]
REPORT FORM: [form name]
```

If subtotals have been requested, the following appears before each new subtotal group:

```
*[field name]:  [field value]
```

The "*" can also be either "**" or "***," depending on the line of the subtotal.

This follows the subtotal group:

```
*TOTAL FOR [field name]:  [field value]
```

The set of totals appears under the appropriate columns. If totals have been requested, the following appears at the end of the report:

```
** GRAND TOTAL **
```

The set of totals appears under the appropriate columns.

7. EXAMPLE OF RECALL SYSTEM

Figures 1 to 9 illustrate the various capabilities of RECALL.

```
:LOAD DCR "RECALL"
:RUN
RECALL SYSTEM -REVISION 8/22/75
DATE(MM/DD/YY)? 8/22/75
COMMAND? CREATE FUZEFILE
PLEASE TYPE IN STRUCTURE OF DATA BASE

NAME; WIDTH; TYPE; DECPL
 1 ? MFR; 5; C
 2 ? LOT; 10; C
 3 ? SIZE; 4; I
 4 ? REJECTS; 7; I
 5 ? REJ. SIZE; 8; N; 2
 6 ?

MFR       ; LOT      ; SIZE      ; REJECTS ; REJ. SIZE;
? RYB; R7001; 100; 2; 0
? HP; HP001; 105; 3; 0
? UCR; UCR3001; 80; 4; 0
? HAM; H1001; 200; 8; 0
? RYB; R7002; 103; 3; 0
? RYB; R7003; 92; 1; 0
? HP; HP002; 103; 2; 0
? UCR; UCR3002; 81; 3; 0
?
 8 RECORDS
COMMAND? LIST

RECNO MFR   LOT       SIZE REJECTS REJ. SIZE

    1 RYB   R7001      100    2     0.00
    2 HP    HP001      105    3     0.00
    3 UCR   UCR3001     80    4     0.00
    4 HAM   H1001      200    8     0.00
    5 RYB   R7002      103    3     0.00
    6 RYB   R7003       92    1     0.00
    7 HP    HP002      103    2     0.00
    8 UCR   UCR3002     81    3     0.00

 8 RECORDS
COMMAND? PRINT MFR; SIZE

MFR   SIZE

RYB   100
HP    105
UCR    80
HAM   200
RYB   103
RYB    92
HP    103
UCR    81

 8 RECORDS
```

Figure 1.   Example of RECALL run for CREATE and LIST commands.

```
COMMAND? 1,3:5LIST MFR

RECNO MFR

    1 RYB
    3 UCR
    4 HAM
    5 RYB

 4 RECORDS
COMMAND? 3:4CHANGE REJECTS,SIZE FOR MFR='UCR'
REJECTS         4? 5
SIZE         80? 84
COMMAND? PRINT LOT,REJECTS FOR REJECTS>3

LOT         REJECTS

UCR3001          5
H1001            8

 2 RECORDS
COMMAND? DELETE FOR MFR='HAM'
COMMAND? REPLACE SIZE*   WITH 2*SIZE,REJ.SIZE &
? WITH (REJECTS/SIZE)*100 FOR REJECTS>0
COMMAND? LIST

RECNO MFR    LOT        SIZE REJECTS REJ.SIZE

    1 RYB    R7001       200      2    1.00
    2 HP     HP001       210      3    1.44
    3 UCR    UCR3001     168      5    2.99
    4 RYB    R7002       206      3    1.44
    5 RYB    R7003       184      1      55
    6 HP     HP002       206      2     .99
    7 UCR    UCR3002     162      3    1.88

 7 RECORDS
```

Figure 2. Example of RECALL run for CHANGE and REPLACE commands.

24

```
COMMAND? SUM REJECTS FOR MFR='RYB'
 REJECTS FOR MFR='RYB'
 6 ;
 3 RECORDS
COMMAND? SUM REJECTS*SIZE
 REJECTS*SIZE
 3570 ;
 7 RECORDS
COMMAND? APPEND

MFR       ; LOT       ; SIZE      ; REJECTS ; REJ. SIZE;
? HP; HP003; 106; 5; 0
? RYB; R7004; 95; 2; 0
? UCR
LOT       ? UCR3003
SIZE      ? 90
REJECTS ? 6
REJ. SIZE? 0
?
 10 RECORDS
COMMAND? REPLACE REJ. SIZE WITH (REJECTS/SIZE)*100 &
? FOR REJ. SIZE=0
COMMAND? LIST

RECNO MFR    LOT       SIZE REJECTS REJ. SIZE

    1 RYB    R7001      200       2    1.00
    2 HP     HP001      210       3    1.44
    3 UCR    UCR3001    168       5    2.99
    4 RYB    R7002      206       3    1.44
    5 RYB    R7003      184       1     .55
    6 HP     HP002      206       2     .99
    7 UCR    UCR3002    162       3    1.88
    8 HP     HP003      106       5    4.77
    9 RYB    R7004       95       2    2.11
   10 UCR    UCR3003     90       6    6.66

 10 RECORDS
```

Figure 3. Example of RECALL run for SUM and APPEND commands.

```
COMMAND? SORT ON MFR; LOT
COMMAND? LIST

RECNO MFR      LOT        SIZE REJECTS REJ. SIZE

      1 HP     HP001      210       3    1.44
      2 HP     HP002      206       2     .99
      3 HP     HP003      106       5    4.77
      4 RYB    R7001      200       2    1.00
      5 RYB    R7002      206       3    1.44
      6 RYB    R7003      184       1     .55
      7 RYB    R7004       95       2    2.11
      8 UCR    UCR3001    168       5    2.99
      9 UCR    UCR3002    162       3    1.88
     10 UCR    UCR3003     90       6    6.66

  10 RECORDS
COMMAND? SAVE TO FUZE

STOP  MOUNT TAPE TO SAVE DATABASE-TYPE CONTINUE
:CONTINUE
COMMAND? REPLACE MFR WITH MFR+'S'
COMMAND? MERGE ON MFR FROM FUZE

STOP MOUNT TAPE AND TYPE CONTINUE
:CONTINUE
COMMAND? LIST MFR

RECNO MFR

      1 HP
      2 HP
      3 HP
      4 HPS
      5 HPS
      6 HPS
      7 RYB
      8 RYB
      9 RYB
     10 RYB
     11 RYBS
     12 RYBS
     13 RYBS
     14 RYBS
     15 UCR
     16 UCR
     17 UCR
     18 UCRS
     19 UCRS
     20 UCRS

  20 RECORDS
```

Figure 4. Example of RECALL run for SORT and MERGE commands.

```
COMMAND? SIZE
 20 RECORDS
COMMAND? DELETE FOR MFR>'I'
COMMAND? LIST .

RECNO MFR     LOT        SIZE REJECTS REJ. SIZE

     1 HP     HP001       210     3      1. 44
     2 HP     HP002       206     2       . 99
     3 HP     HP003       106     5      4. 77
     4 HPS    HP001       210     3      1. 44
     5 HPS    HP002       206     2       . 99
     6 HPS    HP003       106     5      4. 77

  6 RECORDS
COMMAND? APPEND FROM FUZE

STOP MOUNT TAPE AND TYPE CONTINUE
:CONTINUE
COMMAND? LIST MFR

RECNO MFR

     1 HP
     2 HP
     3 HP
     4 HPS
     5 HPS
     6 HPS
     7 HP
     8 HP
     9 HP
    10 RYB
    11 RYB
    12 RYB
    13 RYB
    14 UCR
    15 UCR
    16 UCR

 16 RECORDS
```

Figure 5. Example of RECALL run for SIZE, DELETE, and TAPE APPEND commands.

27

```
COMMAND? STRUCTURE
FIELD TYPE WIDTH  NAME
  1    C     5    MFR
  2    C    10    LOT
  3    I     4    SIZE
  4    I     7    REJECTS
  5    N     8 , 2 REJ. SIZE
COMMAND? 1, 2:3, 10:16 SAVE TO FUZE

STOP  MOUNT TAPE TO SAVE DATABASE-TYPE CONTINUE
:CONTINUE
COMMAND? LOAD FUZE

STOP MOUNT TAPE AND TYPE CONTINUE
:CONTINUE
DATABASE FUZE LAST SAVED 8/22/75 HAS BEEN LOADED.
COMMAND? LIST

RECNO MFR    LOT       SIZE REJECTS REJ. SIZE

    1 HP     HP001      210     3     1. 44
    2 HP     HP002      206     2      . 99
    3 HP     HP003      106     5     4. 77
    4 RYB    R7001      200     2     1. 00
    5 RYB    R7002      206     3     1. 44
    6 RYB    R7003      184     1      . 55
    7 RYB    R7004       95     2     2. 11
    8 UCR    UCR3001    168     5     2. 99
    9 UCR    UCR3002    162     3     1. 88
   10 UCR    UCR3003     90     6     6. 66

  10 RECORDS
COMMAND? COUNT FOR REJECTS>1 AND MFR='RYB'

   3 RECORDS
COMMAND? AVERAGE REJ. SIZE, REJECTS
  REJ. SIZE, REJECTS
  2. 383 ; 3. 2
  10 RECORDS
```

Figure 6.   Example of RECALL run for STRUCTURE, LOAD, COUNT, and
            AVERAGE commands.

```
COMMAND? REPORT

STOP -MOUNT REPORT TAPE AND TYPE CONTINUE
:CONTINUE
REPORT OUTPUT TO? T
REPORT FORM NAME? FUZE RECORD
UPDATE REPORT FORM? YES
HEADING? YES
DOUBLE SPACE? NO
TOTALS? YES
SUBTOTALS? YES
BY ITEMS? MFR
SUMMARY REPORT ONLY? NNO
 1 WIDTH; CONTENTS? 6; MFR
 2 WIDTH; CONTENTS? 10; LOT
 3 WIDTH; CONTENTS? 4; SIZE
 4 WIDTH; CONTENTS? 7; REJECTS
 5 WIDTH; CONTENTS? 8; (REJECTS/SIZE)*100
 6 WIDTH; CONTENTS?
COL HEADING
 1 ? MANU-/FACT
 2 ? LOT NAME
 3 ? SIZE
 4 ? REJECTS
 5 ? %/REJECTS
COLUMNS-TOTALS; NO OF DECIMAL PLACES
 3 ? YES; 0
 4 ? YES; 0
 5 ? NO; 2

STOP SELECT PRINTER AND TYPE CONTINUE
:CONTINUE
```

Figure 7. Example of REPORT command.

```
PAGE 1
DATE: 8/22/75
DATABASE: FUZE
REPORT FORM: FUZE RECORD

MANU-   LOT NAME    SIZE REJECTS %
FACT                         REJECTS


   * MFR      : HP

HP       HP001       210       3      1.44

HP       HP002       206       2       .99

HP       HP003       106       5      4.77

   *TOTAL FOR MFR       : HP
                     522      10

   * MFR      : RYB

RYB      R7001       200       2      1.00

RYB      R7002       206       3      1.44

RYB      R7003       184       1       .55

RYB      R7004        95       2      2.11

   *TOTAL FOR MFR       RYB
                     685       8

   * MFR      : UCR

UCR      UCR3001     168       5      2.99

UCR      UCR3002     162       3      1.88

UCR      UCR3003      90       6      6.66
** GRAND TOTAL **
                    1627      32
```

Figure 8. Example of REPORT command continued.

30

```
COMMAND? REPORT

STOP -MOUNT REPORT TAPE AND TYPE CONTINUE
:CONTINUE
REPORT OUTPUT TO? FILE
REPORT FORM NAME? BLANK
UPDATE REPORT FORM? YES
HEADING? NO
DOUBLE SPACE? NO
TOTALS? NO
  1 WIDTH; CONTENTS? 5; MFR
  2 WIDTH; CONTENTS? 10; SIZE
  3 WIDTH; CONTENTS?
COLUMNS- NO OF DECIMAL PLACES
  2 ? 0

STOP LOAD DATA TAPE AND TYPE CONTINUE
:CONTINUE
COMMAND? CREATE TEST
PLEASE TYPE IN STRUCTURE OF DATA BASE

NAME; WIDTH; TYPE; DECPL
  1 ? MFR; 5; C
  2 ? SIZE; 10; I
  3 ?

MFR       ; SIZE
?
  0 RECORDS
COMMAND? APPEND FROM FILE

STOP MOUNT TAPE AND TYPE CONTINUE
:CONTINUE
COMMAND? LIST

RECNO MFR    SIZE

      1 HP        210
      2 HP        206
      3 HP        106
      4 RYB       200
      5 RYB       206
      6 RYB       184
      7 RYB        95
      8 UCR       168
      9 UCR       162
     10 UCR        90

  10 RECORDS
COMMAND? QUIT

STOP -PROGRAM COMPLETED
```

Figure 9. Example of REPORT command to generate tape.

APPENDIX A.--USING RECALL DATA BASE IN BASIC PROGRAM

APPENDIX A.--SUMMARY

It is possible to take the RECALL data base stored on tape and, with a call to a subroutine, retrieve the fields for each record in the data base. Hence, the user can write his own BASIC programs to manipulate the data base that was originally created using RECALL. The two routines are called DEFFN'81 and DEFFN'82. They appear in lines 9001 to 9011 in the file ACCESS. There is also a dimension statement in line 1. The user loads the file and then adds the appropriate statements to define the operations he wishes to perform.

The user calls DEFFN'81 once to set up the data tape and read in the structured information. The definition of each variable is given in the program listing (fig. A-1). The user then calls DEFFN'82 each time a new record is desired. The system returns the set of fields in the array W8$(). The record counter U8 is initially set to the number of records. When it reaches zero, all the records have been read.

For example, read in the data base that contains numeric information in fields 3 and 4 that should be checked to determine how many records have identical values in the two fields.

```
LOADDCF "ACCESS"

10 REM** I IS COUNT

20 REM  SET UP TAPE:  GOSUB'81

30 IF U8=0 THEN 90

40 REM READ RECORD:  GOSUB'82

50 CONVERT W8$(3) TO X:  CONVERT W8$ (4) TO Y

60 IF X<>Y THEN 30

70 REM FIELDS ARE SAME

80 I=I+1:  GOTO 30

90 PRINT "COUNT ON RECORDS WITH IDENTICAL FIELDS 3 and 4-";I

100 STOP
```

```
1DIM X8$(17)11,W8$(15)32,Y8$(12)20
9001DEFFN'81:REM INITIATE TAPE READ:T8=12:S8=20
9002REM X8$-FILE NAME;X8-# OF 20 CHAR BLOCKS;Y8$-DATE;Y8-# OF  B
LOCKS/RECORD;X8$()-FIELD INFO;Z8-# OF FIELDS IN RECORD
9003REM FIELD=(1-8)-NAME;9-WIDTH IN BIN;10-TYPE;11-DEC PL IN BIN

9004STOP "MOUNT TAPE AND TYPE CONTINUE":DATA LOAD "STRUCT":DATA
LOAD X8$,X8,Y8$,Y8,X8$(),Z8:U8=X8/Y8:DATA LOAD "FILE":RETURN
9005DEFFN'82:REM READ RECORD:S8=20
9006REM T8-BLOCK COUNT;S8-CHAR COUNT;U8-END OF FILE INDICATOR;W8
$()-ARRAY OF FIELDS IN CHAR FORM
9007FOR R8=1TO Z8:Q8=VAL(STR(X8$(R8),9,1)):W8$(R8)=" ":P8=1
9008S8=S8+1:IF S8<21THEN 9009:S8=1:T8=T8+1:IF T8<13THEN 9009:DAT
A LOAD Y8$():T8=1
9009N8=Q8:IF S8+Q8-1<21THEN 9010:N8=21-S8
9010Q8=Q8-N8:STR(W8$(R8),P8,N8)=STR(Y8$(T8),S8,N8):S8=S8+N8-1:P8
=P8+N8:IF Q8>0THEN 9008
9011NEXT R8:U8=U8-1:RETURN
```

Figure A-1. Listing of ACCESS.

APPENDIX B.--LISTING OF SOURCE CODE

APPENDIX B.--SUMMARY

Appendix B gives a brief description of the program developed to implement the RECALL system. The data base is organized as a sequence of 20 byte blocks. Each record has its fields concatenated with no separating characters. However, every record uses an integral number of blocks. Hence, if the actual number of characters in the record is 61, four blocks are used, but if the number is 60, then three blocks are used. There are 12 blocks to a sector, i.e., the space permitted for a Wang physical record (max 253 bytes). At any time, five sectors are in memory and are called a page. The remaining sectors are saved in temporary storage on the removable disk (line 2080 to 2399).

A description of the important variables in the program is given in table B-I. The precedence table used for evaluating conditions is given in table B-II.

The main program is stored in the file RECALL. It handles the general structure of a command and contains all the general-purpose subroutines. The file RETR1 handles the commands QUIT, CREATE, APPEND, LOAD, SAVE, and CHANGE. The file RETR2 handles the commands MERGE and SORT. The file RETR3 handles the commands LIST, PRINT, FAST, STRUCTURE, SIZE, SUM, AVERAGE, COUNT, REPLACE, and DELETE. The file RETR4 handles the command REPORT. The definitions of all the subroutines are given in table B-III. The listing of all the program files appears in figures B-1 to B-5.

TABLE B-I. VARIABLE DEFINITIONS

Name	Meaning
Data base format	
R$(1-60)	Page of blocks
R1	Present block index within page
R0	Present page index in memory
R2	Number of blocks allowed in memory (60)
R3	Last-used character index in present block
R4	Number of sectors per page (five)
Structure file	
F$	File name
F4	Number of blocks in file
F1$	File date
F1	Number of blocks per record in file
F$(1-17)	Field format for record (bytes 1 to 8: name, 9: width in BIN, 10: type (I,C,N), 11: number of decimals in BIN)
F0	Number of fields in record
Range list	
R6	Number of range pairs
R0(1-5)	Starting record index
R1(1-5)	Ending record index
General	
C0	Command index
B$(1-15)	Field values for record
L0	Number of fields in field list
L$15	Field index table for list
C	Count on I/O buffer
C$(1-12)	I/O buffer for records
C1$(1-12)	I/O buffer for records
FOR condition	
D$(1-12)	Conditon string stack
P$(1-80)	Condition operator stack
D0	Number entries in condition stack
D1	Value of condition (1: true, 0: false)
D2	Position of first condition stack entry following FOR
E4	Number of string constants in stack
E5	Number of numerical constants in stack
E2	Number of string expressions less constant in string stack
E1	Number of numerical expressions less constant in number stack
E(1-10)	Condition number stack
T$(1-10)	Temporary terminal stack
E3	Number of terminals in temporary stack

TABLE B-I. VARIABLE DEFINITIONS (Cont'd)

Name	Meaning

SORT-MERGE

Name	Meaning
D$(1-12)	Output buffer to temporary disk file
I8	Count on blocks in D$
B$(1-3)	Field values for first file
B$(6-8)	Field values for second file
E4	Number of pages for command SORT
Q	Number of records in present page to sort
R0(1-2)	Index of last record for files 1 and 2
E(6-7)	Index of record processed for files 1 and 2
E(9-10)	Fetch indicator for files 1 and 2 (1: fetch)
R0(3-4)	Block index for files 1 and 2
E(4-5)	Index of sector for files 1 and 2
G(1-3)	Block index for field in record
E(1-3)	Character position for field in record
R1(1-3)	Length of field

Report variables

Name	Meaning
H$	Data base name
E$	Report form name
G$(1)	Heading indicator
G$(2)	Double space indicator
G$(3)	Totals indicator
G$(4)	Subtotals indicator
G$(5)	Summary report indicator
G(1-5)	Index of subtotal item field
X2$(1-15)	Width of report column
X3$(1-16)	Pointer to first position in condition stack for expression
S$(15,3)	Heading table for columns
H$(1-15)	Report column information (byte 1: type [C,I,N], 2: total [Y,N], 3: number of decimal places in BIN)
T1	Page count
T2	Line count
T3	Number of columns
T4	Number of items
X4(5,1-3)	Column subtotals
X4(5,4)	Column grand total
M$(1-3)	Value of subtotal item field
K9	Pointer to last position in condition stack before column contents information (i.e., last position of FOR condition)

TABLE B-II. PRECEDENCE TABLE FOR
CONDITIONS

Code	Precedence	Name
1	0	Numeric field name
2	0	Numeric constant
3	0	Character field name
4	0	Character constant
5	15→0*	(
6	0	)
7	13	↑
8	12	*
9	12	/
10	11	+
11	11	-
12	10	<
13	10	>
14	10	=
15	10	#
16	10	<=
17	10	>=
18	9	NOT
19	8	AND
20	7	OR
21	10	IN
22	10	NOT IN
23	14	- (Unary)
24	6	WITH
	0	FOR
	0	;

*Precedence is initially 15, but is
stored in the stack as 0.*

APPENDIX B

TABLE B-III.   SUBROUTINE DEFINITIONS

Index	Definition
1	Reads in field from input string
2	Initializes data base
3(C$,N)	Checks for reserved word C$
4	Converts string to number
5(N)	Retrieves page containing record N
6	Loads data base tape
7	Fetches record and checks for condition (I4=1 indicates record is to be used)
8(M)	Stores field in record
9(N)	Retrieves field index from field-list table
10(N)	Stores field index in field-list table
12(M)	Retrieves field in record
13	Retrieves record
14	Stores record
16	Stores FOR condition on stack
19(X1,X2)	Evaluates condition stack over indices X1-X2
20	Stores data-base structure for field
21	Reads and stores data-base field value after validation
23	Reads and stores field list
25	Adds constant or field to evaluation stacks
26	Checks for string in string
28	Retrieves expression as string for updating field
29(Y)	Converts number to string in proper format
31(X)	Saves page onto disk

TABLE B-III. SUBROUTINE DEFINITIONS (Cont'd)

Index	Definition
34	Transfers temporary disk file onto active disk file
37	Saves block onto temporary disk file
40	Selects temporary file for SORT-MERGE
41	Defines location of SORT-MERGE fields within record
42	Fetches field value from active file
43	Fetches field value from tape file
44	Compares active and tape file and performs merge
45(U)	Loads sector for file U
46(U)	Fetches field value from file U
47	Compares field from two files and performs merge
53(X,Y)	Stores field or constant index in conditional stack
54	Checks for field name (X=1 if there is a field name)
63	Updates character count in field retrieval
68	Fetches record to save in I/O buffer after checking FOR condition
69	Fetches record to save in I/O buffer
70	Fetches page from disk
71	Stores page onto disk
91	Prints page heading for report
92(J,K)	prints subtotal item field value
93(K,Y)	Converts numerical column value to string
94(T)	Right justifies numerical string
95	Increments line count
96	Prints subtotal values for numerical columns
97	Prints column values and updates subtotals

```
10COM A$64, B$32, C$32, R$(60)20, C1$3, C2$3, E$(17)11, F$8, E1$8, E2$(4)
5, F3$11, B$(15)32, C$(12)20, G$8, H$8, R0(5), R1(5), C1$(12)20, L$15
20COM T$(10)1, E(10), D$(12)20, P$(80)1, O$(11)1, N$(4)3, Q$(24)1, X$1,
G(5), D1$(12)20
22 COM I, R2, R4, F1, F5, L, I4, R6, F1, F4, C0, I1, F0, R1, R0, X, X1, X2, Y, M, N,
I7, I9, K9, J, C, I6, L0, K, D0, D1, D2, D3, R3, I8, E1, E2, E3, V, I3, I5, E4, E5, U
30DATA "NAME", "WIDTH", "TYPE", "DECPL", "(", ")", "^", "*", "/", "+", "-"
, "<", ">", "=", "#", "NOT", "AND", "OR", "IN"
40FOR I=1TO 4:READ F2$(I):NEXT I:FOR I=1TO 11:READ O$(I):NEXT I:
FOR I=1TO 4:READ N$(I):NEXT I:FOR I=1TO 24:READ X:BIN(Q$(I))=X:N
EXT I:R2=60:R4=5:F1=1
50DATA 0, 0, 0, 0, 15, 0, 13, 12, 12, 11, 11, 10, 10, 10, 10, 10, 10, 9, 8, 7, 10, 10
, 14, 6
60ON ERROR B$, C$GOTO 70:GOTO 80
70PRINT "ERROR "; B$; "-LINE "; C$:GOTO 100
80DATA SAVE DC OPEN R TEMP , 2080, 2399:DATA SAVE DA R(2399, B$)END

90PRINT "RECALL SYSTEM -REVISION 8/22/75": INPUT "DATE(MM/DD/YY)"
, F1$
100D0, E4, E5, R6, D2=0:REM READ COMMAND:INPUT "COMMAND", A$:IF NUM(A
$)=0THEN 160:REM READ RANGES:F5=1
110L=NUM(A$):IF L=0THEN 220:B$=STR(A$, 1, L):GOSUB 4:IF I4=0THEN
220:IF F5=2THEN 150:R6=R6+1:R0(R6)=X:R1(R6)=X
120X$=STR(A$, L+1):A$=STR(A$, L+1):IF X$="; "THEN 130:IF X$<>":"THE
N 170:F5=2
130A$=STR(A$, 2):GOTO 110
150R1(R6)=X:F5=1:GOTO 120
160R6=1:R0(1)=1:R1(1)=F4/F1
170REM READ COMMAND TYPE
180DATA "QUI", "CRE", "APP", "SAV", "LOA", "CHA", "LIS", "PRI", "FAS", "R
EP", "COU", "STR", "SIZ", "SUM", "AVE", "DEL", "MER", "SOR"
190RESTORE 44:A$=STR(A$, POS(A$<>" ")):C1$=STR(A$, 1, 3):C0=0
200 FOR I=1TO 18:READ C2$:IF C2$<>C1$THEN 210:C0=I:I=18
210NEXT I:IF C0<>0THEN 225
220PRINT "ILLEGAL COMMAND":GOTO 100
225 IF C0<>10THEN 230:IF STR(A$, 4, 1)<>"O"THEN 230:C0=19
230A$=STR(A$, POS(A$=" "))
232 IF C0>6THEN 233:IF V=1THEN 245:V=1:DATA SAVE DC CLOSE:LOAD D
C R"RETR1"240, 1150
233 IF C0>16THEN 234:IF V=3THEN 245:V=3:DATA SAVE DC CLOSE:LOAD
DC R"RETR3"240, 1150
234IF C0=19THEN 235:IF V=2THEN 245:V=2:DATA SAVE DC CLOSE:LOAD D
C R"RETR2"240, 1150
235IF V=4THEN 245:V=4:DATA SAVE DC CLOSE:LOAD DC R"RETR4"240, 115
0
240 REM *** START OF COMMANDS
245 REM
1150 REM *** END OF COMMANDS
```

Figure B-1. Listing of RECALL.

```
1160DEFFN'1:REM READ IN NEXT FIELD:IF C1$<>"CHA"THEN 1170:I1=1:B
$=B$(I):RETURN
1170IF A$="&"THEN 1175:B$=A$:I1=POS(A$=",")":IF I1=0THEN 1180:IF
I1=1THEN 1190:B$=STR(B$,1,I1-1):GOTO 1200
1175INPUT A$:GOTO 1170
1180A$=" ":IF B$=" "THEN 1210:I1=1:GOTO 1210
1190B$=" "
1200A$=STR(A$,I1+1):I1=1
1210RETURN
1220DEFFN'2,F0,R1,R3,R0,F4=0:DBACKSPACE BEG :L=POS(A$<>" "):IF L
=0THEN 1230:F$=STR(A$,L):RETURN
1230INPUT "DATABASE",F$:RETURN
1240DEFFN'3(C$,N)
1250IF LEN(A$)<NTHEN 1260:IF STR(A$,1,N)=C$THEN 1270:A$=STR(A$,2
):GOTO 1250
1260N=0:RETURN
1270A$=STR(A$,N+1):A$=STR(A$,POS(A$<>" ")):RETURN
1280DEFFN'4:REM CONVERT TO NUMBER
1290I4=0:IF B$=" "THEN 1300:IF NUM(B$)=0THEN 1300:IF NUM(B$)<LE
N(B$)THEN 1300:CONVERT B$TO X:I4=1:GOTO 1310
1300PRINT "ILLEGAL NUMBER"
1310RETURN
1320DEFFN'6:STOP "MOUNT TAPE AND TYPE CONTINUE":IF C0=3THEN 1325
:DATA LOAD "STRUCT":DATA LOAD G$,I9,H$,F1,F$(),F0:GOTO 1327
1325DATA LOAD "STRUCT":DATA LOAD G$,I9
1327IF A$=G$THEN 1330:PRINT "WRONG DATABASE ",G$:I9=0:RETURN
1330DATA LOAD "FILE":RETURN
1340DEFFN'8(M): REM STORE FIELD IN RECORD:K=1
1350R3=R3+1:IF R3<21THEN 1370:R1=R1+1:R3=1:IF R1<=R2THEN 1370:GO
SUB '71:R0=R0+1:R1=1
1370GOSUB '63:STR(R$(R1),R3,I8)=STR(B$,K,I8):K=K+I8:R3=R3+I8-1:I
F M>0THEN 1350:RETURN
1400DEFFN'12(M):REM RETRIEVE FIELD IN RECORD:B$=" ":K=1
1410R3=R3+1:IF R3<21THEN 1440:R3=1:R1=R1+1:IF R1<=R2THEN 1420:GO
SUB '70:R0=R0+1:R1=1
1420C=C+1:IF C>12THEN 1430:C$(C)=R$(R1):GOTO 1440
1430C1$(C-12)=R$(R1)
1440GOSUB '63:STR(B$,K,I8)=STR(R$(R1),R3,I8):R3=R3+I8-1:K=K+I8:I
F M>0THEN 1410:RETURN
1470DEFFN'13:REM RETRIEVE RECORD
1480FOR I7=1TO F0:GOSUB '12(VAL(STR(F$(I7),9))):B$(I7)=B$:NEXT I
7:RETURN
1490DEFFN'14:REM STORE RECORD
1500FOR I7=1TO F0:B$=B$(I7):GOSUB '8(VAL(STR(F$(I7),9))):NEXT I7
:RETURN
1510DEFFN'5(N):REM INITIATE FETCH:N=(N-1)*F1+1
1520F5=INT((N-1)/R2):R1=N-F5*R2-1:DBACKSPACE BEG  IF N<1THEN 152
5:DSKIP F5*R4S:GOSUB '70:R0=F5:GOTO 1530
1525R0=0:R1=-F1
1530RETURN
1540DEFFN'9(N):L=VAL(STR(L$,N)):RETURN
1550DEFFN'10(N):BIN(STR(L$,N))=I:RETURN
```

Figure B-1. Listing of RECALL (Cont'd).

```
1560DEFFN'16:REM SET UP FOR COND:E3,I6=0
1570IF A$<>" "THEN 1580:I6=1:GOTO 1800
1580B$=STR(A$,1,1):IF B$<>" "THEN 1590:A$=STR(A$,2):GOTO 1580
1590IF B$<>"&"THEN 1600:INPUT A$:GOTO 1570
1600IF B$<>";"THEN 1620:A$=STR(A$,2)
1610IF E3=0THEN 1570:FOR I=E3TO 1STEP -1:D0=D0+1:P$(D0)=T$(I):NE
XT I:E3=0:GOTO 1570
1620FOR I=1TO 11:IF B$<>O$(I)THEN 1640:K=I+4:IF (K-12)*(K-13)<>0
THEN 1630:IF STR(A$,2,1)<>"="THEN 1630:K=K+4:A$=STR(A$,2)
1630A$=STR(A$,2):GOTO 1800
1640NEXT I:IF B$<>"'"THEN 1660:E4=E4+1:A$=STR(A$,2):L=POS(A$="'"
):IF L<>0THEN 1650:I6=2:GOTO 1800
1650D$(E4)=STR(A$,1,L-1):A$=STR(A$,L+1):GOSUB '53(4,E4):GOTO 157
0
1660B$=STR(A$,1,3):FOR I=1TO 4:IF B$<>N$(I)THEN 1680:K=I+17:IF S
TR(A$,5,2)<>"IN"THEN 1670:A$=STR(A$,2):K=22
1670A$=STR(A$,4):GOTO 1800
1680NEXT I:L=NUM(A$):IF L=0THEN 1700:IF L<LEN(A$)THEN 1690:L=LEN
(A$)
1690E5=E5+1:CONVERT STR(A$,1,L)TO E(E5):GOSUB '53(2,E5):A$=STR(A
$,L+1):GOTO 1570
1700B$=A$:I9=0
1710X$=STR(A$,1,1):GOSUB '54:IF X=0THEN 1720:I9=I9+1:A$=STR(A$,2
):GOTO 1710
1720IF I9=0THEN 1790:B$=STR(B$,1,I9)
1730IF B$<>"FOR"THEN 1740:D2=D0+E3+1:GOTO 1610
1740IF B$<>"FROM"THEN 1750:D2=D0+1:A$=STR(A$,POS(A$<>" ")):RETUR
N
1750IF B$<>"WITH"THEN 1760:K=24:GOTO 1840
1760FOR I=1TO F0:IF B$<>STR(F$(I),1,8)THEN 1780:X=3:IF STR(F$(I)
,10)="C"THEN 1770:X=1
1770GOSUB '53(X,I):GOTO 1570
1780NEXT I
1790I6=2
1800IF I6<>1THEN 1830:IF E3<>0THEN 1810:RETURN
1810FOR I9=E3TO 1STEP -1:D0=D0+1:P$(D0)=T$(I9):IF VAL(T$(I9))<>5
THEN 1820:I6=2:RETURN
1820NEXT I9:RETURN
1830IF I6<>2THEN 1840:RETURN
1840IF E3<>0THEN 1870
1850IF E3=0THEN 1860:IF K<>11THEN 1860:X=VAL(O$(VAL(T$(E3)))):K=
23:IF X<6THEN 1860:IF X>11THEN 1860:K=11
1860E3=E3+1:BIN(T$(E3))=K:GOTO 1570
1870IF VAL(T$(E3))<>5THEN 1880:IF K<>6THEN 1850:E3=E3-1:GOTO 157
0
1880IF VAL(O$(VAL(T$(E3))))<VAL(O$(K))THEN 1850:D0=D0+1:P$(D0)=T
$(E3):E3=E3-1:GOTO 1840
1890DEFFN'53(X,Y):D0=D0+1:BIN(P$(D0))=X:D0=D0+1:BIN(P$(D0))=Y:RE
TURN
1900DEFFN'54:X=0:IF X$=HEX(2E)THEN 1910:IF X$<HEX(30)THEN 1920:I
F X$>HEX(5A)THEN 1920:IF X$>HEX(3F)THEN 1910:IF X$>=HEX(3A)THEN
1920
1910X=1
1920RETURN
```

Figure B-1. Listing of RECALL (Cont'd).

```
1930DEFFN´25:IF I7>2THEN 1940:E1=E1+1:GOTO 1950
1940E2=E2+1
1950I5=I5+1:X=VAL(P$(I5)):RETURN
1960DEFFN´26:FOR I7=1TO LEN(B$)-LEN(A$)+1:IF A$<>STR(B$,I7,LEN(A
$))THEN 1970:I7=1:RETURN
1970NEXT I7:I7=0:RETURN
1980DEFFN´19(X1,X2):D3=0:E2=E4:E1=E5
1990FOR I5=X1TO X2:I7=VAL(P$(I5)):IF I7>4THEN 2040:GOSUB ´25:ON
I7GOTO 2000,2010,2020,2030
2000CONVERT B$(X)TO E(E1) GOTO 2400
2010E(E1)=E(X):GOTO 2400
2020D$(E2)=B$(X):GOTO 2400
2030D$(E2)=D$(X):GOTO 2400
2040IF E1=E5THEN 2060:IF E1<2THEN 2050:X=E(E1-1)
2050Y=E(E1)
2060IF E2=E4THEN 2080:IF E2<2THEN 2070:A$=D$(E2-1)
2070B$=D$(E2)
2080ON I7-6 GOTO 2090,2100,2110,2120,2140,2150,2170,2190,2210,22
40,2250,2270,2100,2280,2290,2300,2310,2320
2090X=X^Y:GOTO 2390
2100X=X+Y:GOTO 2390
2110X=X/Y:GOTO 2390
2120IF E2<>E4THEN 2130:X=X+Y:GOTO 2390
2130STR(D$(E2-1),LEN(A$)+1)=B$:E2=E2-1:GOTO 2400
2140X=X-Y:GOTO 2390
2150IF E2<>E4THEN 2160:IF X<YTHEN 2360:GOTO 2370
2160IF A$<B$THEN 2330:GOTO 2340
2170IF E2<>E4THEN 2180:IF X>YTHEN 2360:GOTO 2370
2180IF A$>B$THEN 2330:GOTO 2340
2190IF E2<>E4THEN 2200:IF X=YTHEN 2360:GOTO 2370
2200IF A$=B$THEN 2330:GOTO 2340
2210IF E2<>E4THEN 2220:IF X<>YTHEN 2360:GOTO 2370
2220IF A$<>B$THEN 2330:GOTO 2340
2230IF E2<>E4THEN 2240:IF X<=YTHEN 2360:GOTO 2370
2240IF A$<=B$THEN 2330:GOTO 2340
2250IF E2<>E4THEN 2260:IF X>=YTHEN 2360:GOTO 2370
2260IF A$>=B$THEN 2330:GOTO 2340
2270E(E1)=1-E(E1):GOTO 2400
2280IF X+Y<>0THEN 2360:GOTO 2370
2290GOSUB ´26:IF I7=1THEN 2330:GOTO 2340
2300GOSUB ´26:IF I7=1THEN 2340:GOTO 2330
2310E(E1)=-E(E1):GOTO 2400
2320GOSUB ´28:D3=I5:GOTO 2400
2330E(E1+1)=1:GOTO 2350
2340E(E1+1)=0
2350E1=E1+1:E2=E2-2:GOTO 2400
2360E(E1-1)=1:GOTO 2380
2370E(E1-1)=0
2380E1=E1-1:GOTO 2400
2390E1=E1-1:E(E1)=X
2400NEXT I5
2410IF E1<>E5THEN 2420:RETURN
2420D1=E(E1):RETURN
```

Figure B-1. Listing of RECALL (Cont'd).

```
2430DEFFN'7:I4=1:R3=20:GOSUB '13:IF D2=0THEN 2440:GOSUB '19(D2,D
0):IF D1=1THEN 2440:I4=0
2440RETURN
2450DEFFN'23:I6=0
2460IF A$=" "THEN 2480:GOSUB '16:IF I6=2THEN 2490:IF D0*(D2-1)=0
THEN 2480:X=D0:IF D2=0THEN 2470:X=D2-1
2470FOR L0=1TO X/2:I=VAL(P$(2*L0)):GOSUB '10(L0):NEXT L0:RETURN

2480FOR I=1TO F0:GOSUB '10(I):NEXT I:L0=F0
2490RETURN
2500DEFFN'28:I6=VAL(P$(D3+2)):N=0:IF STR(F$(I6),10,1)<>"C"THEN 2
510:B$=D$(E2):E2=E4:GOTO 2518
2510IF STR(F$(I6),10,1)="I"THEN 2515:N=VAL(STR(F$(I6),11))
2515GOSUB '29(E(E1))
2517C$=" ":STR(C$,1+VAL(STR(F$(I6),9))-LEN(B$))=B$:B$=C$
2518B$(I6)=B$:RETURN
2520DEFFN'29(Y):M1=INT(Y*10^N):CONVERT M1TO B$,(###########):E1
=E5
2530IF STR(B$,1,1)<>"0"THEN 2540:IF LEN(B$)<N+1THEN 2540:B$=STR(
B$,2):GOTO 2530
2540IF M1>=0THEN 2550:T=LEN(B$):STR(B$,2)=B$:STR(B$,1,1)="-":B$=
STR(B$,1,T+1)
2550IF N=0THEN 2560:T=LEN(B$):STR(B$,T-N+2)=STR(B$,T-N+1):STR(B$
,T-N+1,1)=".":B$=STR(B$,1,T+1)
2560RETURN
2580DEFFN'31(X):IF F1*J-INT(F1*J/R2)*R2=0THEN 2590:IF J<>R1(X)TH
EN 2600
2590DBACKSPACE R2/12S:GOSUB '71
2600RETURN
2610 DEFFN'34:DATA SAVE DC #2,END :DATA SAVE DC CLOSE#2:DATA LOA
D DC OPEN F#2,TEMP ,2080,2399:DBACKSPACE BEG
2620DATA LOAD DC #2,C$():IF END THEN 2630:DATA SAVE DC C$():GOTO
2620
2630 DATA SAVE DC CLOSE#2:F4=I9*F1:RETURN
2640DEFFN'37:I9=I9+1:IF C<12THEN 2650:DATA SAVE DC #2,C$():FOR I
5=1TO 12:C$(I5)=C1$(I5):NEXT I5:C=C-12
2650RETURN
2660DEFFN'63:I8=M:IF R3+M-1<21THEN 2670:I8=21-R3
2670M=M-I8:RETURN
2680DEFFN'70:FOR Z=1TO R4:DATA LOAD DC D1$():FOR W=1TO 12:R$((Z-
1)*12+W)=D1$(W):NEXT W:NEXT Z:RETURN
2690DEFFN'71:FOR Z=1TO R4:FOR W=1TO 12:D1$(W)=R$((Z-1)*12+W):NEX
T W:DATA SAVE DC D1$():NEXT Z:RETURN
```

Figure B-1. Listing of RECALL (Cont'd).

```
240DATA LOAD DC OPEN RTEMP ,2080,2399
245ON C0 GOTO 250,260,630,750,720,840
250REM **QUIT:DATA SAVE DC CLOSE:STOP "-PROGRAM COMPLETED"
260REM **CREATE :GOSUB '2
270PRINT "PLEASE TYPE IN STRUCTURE OF DATA BASE":PRINT :FOR I=1T
O 3:PRINT F2$(I);" ";:NEXT I:PRINT F2$(4)
280I=0:PRINT F0+1;:INPUT A$
290I=I+1:GOSUB '20:ON I1+1 GOTO 300,290,310,330
300IF I=1THEN 340
310PRINT F2$(I);:INPUT A$:GOSUB '20:ON I1+1GOTO 310,320,310,330
320I=I+1:GOTO 310
330F0=F0+1:F$(F0)=F3$:F5=F5+VAL(STR(F3$,9)):GOTO 280
340F1=INT((F5-1)*.05)+1
350F5=R1:PRINT :FOR I=1TO F0:PRINT STR(F$(I),1,8);" ";:NEXT I:PR
INT
360R3=20:I=0:INPUT A$
370I=I+1:GOSUB '21:ON I1+1GOTO 380,370,390,360
380IF I=1THEN 410
390PRINT STR(F$(I),1,8);:INPUT A$:GOSUB '21:ON I1+1GOTO 390,400,
390,360
400I=I+1:GOTO 390
410F4=R0*R2+R1:PRINT F4/F1;"RECORDS ":GOSUB '71:GOTO 100
420DEFFN'20
430GOSUB '71:IF I1=0THEN 510:ON IGOTO 440,460,480,460
440IF LEN(B$)<9THEN 450:PRINT "FIELD TOO LONG":I1=2:GOTO 510
450STR(F3$,1,8)=B$:GOTO 510
460GOSUB '4:IF I4=1THEN 470:I1=2:GOTO 510
470BIN(STR(F3$,7+I))=X:IF I=2THEN 510:I1=3:GOTO 510
480IF B$="C"THEN 490:IF B$="I"THEN 490:IF B$="N"THEN 500:PRINT "
ILLEGAL TYPE":I1=2:GOTO 510
490I1=3
500STR(F3$,10)=B$
510RETURN
520DEFFN'21
530GOSUB '1:IF I1=0THEN 620:M=VAL(STR(F$(I),9)):IF B$=" "THEN 61
0
540IF STR(F$(I),10,1)="N"THEN 580:IF M>=LEN(B$)THEN 550:PRINT "F
IELD TOO LONG":I1=2:GOTO 620
550IF STR(F$(I),10,1)="C"THEN 610
560IF NUM(B$)>=LEN(B$)THEN 570:PRINT "ILLEGAL NUMBER":I1=2:GOTO
620
570C$=" ":STR(C$,1+M-LEN(B$))=B$:B$=C$:GOTO 610
580N=VAL(STR(F$(I),11)):L=POS(B$=".") :IF L<>0THEN 590:STR(B$,LEN
(B$)+1)=". ":L=LEN(B$):GOTO 600
590I7=LEN(STR(B$,L+1,LEN(B$)-L)):IF I7=NTHEN 560:IF I7<NTHEN 600
:B$=STR(B$,1,LEN(B$)-1):GOTO 590
600STR(B$,LEN(B$)+1)="0":GOTO 590
610REM STORE FIELD:GOSUB '8(M):IF I<>F0THEN 620:I1=3
620RETURN
```

Figure B-2. Listing of RETR1.

49

APPENDIX B

```
630REM **APPEND
640GOSUB '5(F4/F1):DBACKSPACE R2/125:R1=R1+F1:IF A$=" "THEN 350:
GOSUB '3("FROM",4):IF N=0THEN 220:GOSUB  6:IF I9=0THEN 220
650DATA LOAD C$():IF END THEN 710
670FOR J=1TO 12
680R1=R1+1:IF R1>R2THEN 690:R$(R1)=C$(J):GOTO 700
690GOSUB '71:R0=R0+1:R1=0:GOTO 680
700NEXT J:GOTO 650
710F4=F4+I9:GOSUB '71:REWIND :GOTO 100
720REM **LOAD
730GOSUB '2:A$=F$:GOSUB '6:IF I9=0THEN 220
740PRINT "DATABASE ";G$;" LAST SAVED ";H$;" HAS BEEN LOADED.":F4
=0:GOTO 650
750REM **SAVE
760GOSUB '3("TO",2):IF N=0THEN 220
770G$=STR(A$,1,POS(A$=" ")):A$=STR(A$,POS(A$=" ")):GOSUB '16:IF
I6=2THEN 220
780STOP " MOUNT TAPE TO SAVE DATABASE-TYPE CONTINUE":REWIND :DAT
A SAVE OPEN "STRUCT":DATA SAVE G$,F4,F1$,F1,F$(),F0:DATA SAVE EN
D :DATA SAVE OPEN "FILE":C=0
790I9=0:FOR I=1TO R6:GOSUB '5(R0(I))
800FOR J=R0(I)TO R1(I):GOSUB '68:IF I4=1THEN 810:C=C-F1:GOTO 820
810I9=I9+F1:IF C<=12THEN 820:DATA SAVE C$():FOR K=1TO 12:C$(K)=C
1$(K):NEXT K:C=C-12
820NEXT J:NEXT I:IF C=0THEN 830:DATA SAVE C$()
830DATA SAVE END :REWIND :DATA LOAD "STRUCT":DATA RESAVE G$,I9,F
1$,F1,F$(),F0:REWIND :GOTO 100
840REM **CHANGE:GOSUB '23:IF I6=2THEN 220:FOR I2=1TO R6:GOSUB '5
(R0(I2)):FOR J=R0(I2)TO R1(I2):C=0:GOSUB '7:IF I4=0THEN 860
850FOR I4=1TO L0:GOSUB '9(I4):PRINT STR(F$(L),1,8);" ";B$(L);:B$
(L)=" ":INPUT B$(L):NEXT I4:R3=20:R1=R1-F1:FOR I=1TO F0:GOSUB '2
1:NEXT I
860GOSUB '31(I2):NEXT J:NEXT I2:GOTO 100
1081DEFFN'69:FOR I8=1TO F1
1082R1=R1+1:IF R1<=R2THEN 1083:GOSUB '70:R0=R0+1:R1=1
1083C=C+1:IF C>12THEN 1084:C$(C)=R$(R1):GOTO 1085
1084C1$(C-12)=R$(R1)
1085NEXT I8:RETURN
1086DEFFN'68:I4=1:IF D2=0THEN 1087:GOSUB '7:RETURN
1087GOSUB '69
1088RETURN
```

Figure B-2. Listing of RETR1 (Cont'd).

```
240DATA LOAD DC OPEN RTEMP ,2080,2399
2450N C0-16GOTO 250,500
250REM **MERGE:GOSUB '3("ON",2):IF N=0THEN 220:GOSUB '23:IF I6=2
THEN 220:IF D2<>D0+1THEN 220:GOSUB '6:IF I9=0THEN 220:GOSUB '40:
GOSUB '41
260DATA LOAD C$():DATA LOAD C1$():GOSUB '5(1):I1=F4/F1:I2=I9/F1:
X1,X2=1:R1,I7=1-F1:I8=0
270IF (I1+1)*X1=0THEN 280:GOSUB '42
280IF (I2+1)*X2=0THEN 290:GOSUB '43
290IF I1+I2+2=0THEN 300:GOSUB '44:GOTO 270
300I9=(F4+I9)/F1:DATA SAVE DC #2,D$():GOSUB '34:REWIND :GOTO 100

310DEFFN'40:SELECT #2310:DATA SAVE DC OPEN F#2,TEMP ,2080,2399:R
ETURN
320DEFFN'41:IF L0<3THEN 330:L0=3
330FOR I=1TO L0:GOSUB '9(I):X=1:IF L=1THEN 340:FOR J=1TO L-1:X=X
+VAL(STR(F$(J),9)):NEXT J
340E(I)=X-1-INT((X-1)*.05)*20:G(I)=INT((X-1)*.05):R1(I)=VAL(STR(
F$(L),9)):NEXT I:RETURN
350DEFFN'42:I1=I1-1:IF I1<0THEN 375:R1=R1+F1:IF R1<=R2THEN 360:G
OSUB '70:R1=1
360FOR I=1TO L0:K=R1+G(I):R3=E(I):B$(I)=" ":M=R1(I):J=1
365R3=R3+1:IF R3<21THEN 370:R3=1:K=K+1
370L=M:IF R3+M-1<21THEN 373:L=21-R3
373M=M-L:STR(B$(I),J,L)=STR(R$(K),R3,L):R3=R3+L-1:J=J+L:IF M>0TH
EN 365:NEXT I:RETURN
375B$(1)="^":RETURN
380DEFFN'43:I2=I2-1:IF I2<0THEN 425:I7=I7+F1:IF I7<13THEN 390:FO
R I=1TO 12:C$(I)=C1$(I):NEXT I:DATA LOAD C1$():I7=I7-12
390FOR I=1TO L0:K=I7+G(I):R3=E(I):B$(I+5)=" ":M=R1(I):J=1
395R3=R3+1:IF R3<21THEN 400:R3=1:K=K+1
400L=M:IF R3+M-1<21THEN 405:L=21-R3
405M=M-L:IF K<13THEN 410:STR(B$(I+5),J,L)=STR(C1$(K-12),R3,L):GO
TO 420
410STR(B$(I+5),J,L)=STR(C$(K),R3,L)
420R3=R3+L-1:J=J+L:IF M>0THEN 395:NEXT I:RETURN
425B$(6)="^":RETURN
430DEFFN'44:FOR I=1TO L0:IF B$(I)>B$(I+5)THEN 460:IF B$(I)<B$(I+
5)THEN 440:NEXT I
440X1=1:X2=0:FOR J=R1TO R1+F1-1:I8=I8+1:IF I8<13THEN 450:DATA SA
VE DC #2,D$():I8=1
450D$(I8)=R$(J):NEXT J:RETURN
460X1=0:X2=1:FOR J=I7TO I7+F1-1:I8=I8+1:IF I8<13THEN 470:DATA SA
VE DC #2,D$():I8=1
470IF J<13THEN 480:D$(I8)=C1$(J-12):GOTO 490
480D$(I8)=C$(J)
490NEXT J:RETURN
```

Figure B-3. Listing of RETR2.

51

```
500REM **SORT:GOSUB '3("ON",2):IF N=0THEN 220:GOSUB  23:IF I6=2T
HEN 220:GOSUB '41:E4=INT((F4-1)/R2)+1:Q=R2/F1
510FOR I9=1TO E4:IF F4>=I9*R2THEN 520:Q=(F4-(I9-1)*R2)/F1
520GOSUB '5((I9-1)*(R2/F1)+1):IF Q=1THEN 565:I1=Q+Q:FOR R=1TO Q-
1:R1=1+(R-2)*F1:T=R:GOSUB '42:FOR X=1TO L0:B$(5+X)=B$(X):NEXT X
530FOR X=R+1TO Q:GOSUB '42:FOR X1=1TO L0:IF B$(X1)=B$(X1+5)THEN
540:IF B$(X1)>B$(X1+5)THEN 535:T=X:FOR X2=1TO L0:B$(X2+5)=B$(X2)
:NEXT X2
535X1=L0
540NEXT X1:NEXT X:IF T=RTHEN 560:FOR X1=1TO F1:C$(X1)=R$((R-1)*F
1+X1)
550R$((R-1)*F1+X1)=R$((T-1)*F1+X1):R$((T-1)*F1+X1)=C$(X1):NEXT X
1
560NEXT R
565DBACKSPACE R4S:GOSUB '71:NEXT I9
570IF E4=1THEN 100:X=2
580I1,I2=X*R2/(F1*2):GOSUB '40:I8=0:Z=-I1:FOR X1=1TO E4STEP X:Z=
Z+2*I1:E(7)=Z:E3=F4/F1-E(7):IF E3>0THEN 590:X1=E4:GOTO 660
590IF E3>I2THEN 600:I2=E3
600E(6)=E(7)-I1:R0(1)=E(6)+I1:R0(2)=E(7)+I2:R0(3),R0(4)=1-F1:E(4
)=E(6)*F1/12:E(5)=E(7)*F1/12:GOSUB '45(1):GOSUB '45(2)
610FOR I=1TO 12:FOR J=1TO 2:R$((J-1)*12+I)=R$((J+1)*12+I):NEXT J
:NEXT I:GOSUB '45(1):GOSUB '45(2):E(9),E(10)=1
620IF (R0(1)-E(6)+1)*E(9)=0THEN 630:GOSUB '46(1)
630IF (R0(2)-E(7)+1)*E(10)=0THEN 640:GOSUB '46(2)
640IF R0(1)-E(6)+R0(2)+2-E(7)=0THEN 650:GOSUB '47:GOTO 620
650DATA SAVE DC #2,D$():I8=0:GOTO 665
660I3=(E(7)-I1)*F1:E(4)=I3/12:FOR P=I3+1TO F4STEP 12:GOSUB '45(1
):DATA SAVE DC #2,C$():NEXT P
665NEXT X1
670I9=F4/F1:GOSUB '34:X=X*2:IF X<E4*2THEN 580:GOTO 100
680DEFFN'45(U):DBACKSPACE BEG :DSKIP E(3+U)S:E(3+U)=E(3+U)+1:DAT
A LOAD DC C$():FOR I=1TO 12:R$((U+1)*12+I)=C$(I):NEXT I:RETURN
700DEFFN'46(U):R0(2+U)=R0(2+U)+F1:E(U+5)=E(U+5)+1:IF E(U+5)>R0(U
)THEN 742:IF R0(2+U)<13THEN 710:FOR I=1TO 12:R$((U-1)*12+I)=R$((
U+1)*12+I):NEXT I:GOSUB '45(U):R0(2+U)=R0(2+U)-12
710FOR I=1TO L0:K=R0(2+U)+G(I):R3=E(I):P=I+5*(U-1):B$(P)=" ":M=R
1(I):J=1
715R3=R3+1:IF R3<21THEN 720:R3=1:K=K+1
720L=M:IF R3+M-1<21THEN 725:L=21-R3
725M=M-L:IF K<13THEN 730:STR(B$(P),J,L)=STR(R$(U*12+K),R3,L):GOT
O 740
730STR(B$(P),J,L)=STR(R$((U-1)*12+K),R3,L)
740R3=R3+L-1:J=J+L:IF M>0THEN 715:NEXT I:RETURN
742B$((U-1)*5+1)="^":RETURN
750DEFFN'47:FOR I=1TO L0:IF B$(I)>B$(I+5)THEN 770:IF B$(I)<B$(I+
5)THEN 760:NEXT I
760U=1:GOTO 780
770U=2
780E(10)=U-1:E(9)=2-U:FOR J=R0(U+2)TO R0(U+2)+F1-1:I8=I8+1:IF I8
<13THEN 790:DATA SAVE DC #2,D$():I8=1
790IF J<13THEN 800:D$(I8)=R$(U*12+J):GOTO 810
800D$(I8)=R$((U-1)*12+J)
810NEXT J:RETURN
```

Figure B-3.   Listing of RETR2 (Cont'd)

```
240DATA LOAD DC OPEN RTEMP ,2080,2399
2450N C0-6 GOTO 870,870,870,1060,870,970,990,1000,1000,1090
870REM **LIST,PRINT, OR FAST:I3=0:GOSUB '23:IF I6=2THEN 220:IF C
0=11THEN 910
880IF C0=9THEN 910:J=0:PRINT :IF C0=8THEN 890:PRINT "RECNO";:J=6
:PRINT TAB(6);
890FOR I=1TO '.0:GOSUB '9(I):B$=STR(F$(L),1,8):K=VAL(STR(F$(L),9)
):J=J+K+1:IF LEN(B$)<=KTHEN 900:B$=STR(B$,1,K)
900PRINT B$;TAB(J);:NEXT I:PRINT :PRINT
910FOR I=1TO R6:GOSUB '5(R0(I)):FOR J=R0(I)TO R1(I):GOSUB '7:C=0
:IF I4=0THEN 960:IF C0=11THEN 950
920I9=0:IF C0<>7THEN 940:PRINTUSING 930,J;:I9=6:PRINT TAB(6);
930%#####
940FOR K=1TO L0:GOSUB '9(K):PRINT B$(L);:I9=I9+1+VAL(STR(F$(L),9
)):PRINT TAB(I9);:NEXT K:PRINT
950I3=I3+1
960NEXT J:NEXT I:PRINT :PRINT I3;"RECORDS":GOTO 100
970REM **STRUCTURE:PRINT "FIELD";" TYPE";" WIDTH";"  NAME":FOR I
=1TO F0:PRINT I;TAB(6);STR(F$(I),10,1);TAB(11);VAL(STR(F$(I),9))
;TAB(14);:IF STR(F$(I),10,1)<>"N"THEN 980:PRINT ",";VAL(STR(F$(I
),11));
980PRINT TAB(18);STR(F$(I),1,8):NEXT I:GOTO 100
990REM **SIZE:PRINT F4/F1;"RECORDS":GOTO 100
1000REM **SUM OR AVERAGE:FOR I=1TO 5:G(I)=0:NEXT I:PRINT A$:I3=0
:GOSUB '16:IF I6=2THEN 220:T=D0:IF D2=0THEN 1010:T=D2-1
1010FOR I=1TO R6:GOSUB '5(R0(I)):FOR J=R0(I)TO R1(I):C=0:GOSUB
7:IF I4=0THEN 1030:GOSUB '19(1,T):U=E1
1020I3=I3+1:FOR K=1+E5TO E1:G(K)=G(K)+E(K):NEXT K
1030NEXT J:NEXT I:IF I3=0THEN 1050:IF C1$<>"AVE"THEN 1040:FOR K=
1+E5TO U:G(K)=G(K)/I3:NEXT K
1040FOR I=E5+1TO U:PRINT G(I);" ";:NEXT I:PRINT
1050PRINT I3;"RECORDS":GOTO 100
1060REM **REPLACE:GOSUB '16:IF I6=2THEN 220:FOR I=1TO R6:GOSUB '
5(R0(I)):FOR J=R0(I)TO R1(I):C=0:GOSUB '7:IF I4=0THEN 1080:X=D0:
IF D2=0THEN 1070:X=D2-1
1070GOSUB '19(1,X):R3=20:R1=R1-F1:GOSUB '14
1080GOSUB '31(I):NEXT J:NEXT I:GOTO 100
1081DEFFN'69:FOR I8=1TO F1
1082R1=R1+1:IF R1<=F2THEN 1083:GOSUB '70:R0=R0+1:R1=1
1083C=C+1:IF C>12THEN 1084:C$(C)=R$(R1):GOTO 1085
1084C1$(C-12)=R$(R1)
1085NEXT I8:RETURN
1086DEFFN'68:I4=1:IF D2=0THEN 1087:GOSUB '7:GOTO 1088
1087GOSUB '69
1088RETURN
1090REM **DELETE:GOSUB '16:IF I6=2THEN 220:C,I9=0:SELECT #2 310:
DATA SAVE DC OPEN F#2,TEMP ,2080,2399:GOSUB '5(1):I=1:FOR U=1TO
F4/F1:IF I>R6THEN 1100:IF U>=R0(I)THEN 1110
1100 GOSUB '68:GOSUB '37:GOTO 1140
1110FOR J=R0(I)TO R1(I):GOSUB '68:IF I4=0THEN 1120:C=C-F1:GOTO 1
130
1120GOSUB '37
1130NEXT J:U=U+R1(I)-R0(I):I=I+1
1140NEXT U:IF C=0THEN 1150:DATA SAVE DC #2,C$()
1150GOSUB '34:GOTO 100
```

Figure B-4. Listing of RETR3.

```
240DATA LOAD DC OPEN RTEMP ,2080,2399
245DIM G$(5)1,X2$(15)1,X3$(16)1,H$(15)3,S$(15,3)10,M$(3)20,X5$(3
)3,X4(5,4)
250REM **REPORT:GOSUB '16:IF I6=2THEN 220:K9=D0:STOP "-MOUNT REP
ORT TAPE AND TYPE CONTINUE":C1$="Y"
260H$=" ":INPUT "REPORT OUTPUT TO",H$:E$=" " INPUT "REPORT FORM
NAME",E$:IF E$=" "THEN 280
270INPUT "UPDATE REPORT FORM",C1$:C1$=STR(C1$,1,1):IF C1$="Y"THE
N 280:DATA LOAD "REPORT":DATA LOAD G$(),G(),T4:GOTO 390
280INPUT "HEADING",G$(1):INPUT "DOUBLE SPACE",G$(2):INPUT "TOTAL
S",G$(3):IF G$(3)="N"THEN 380:INPUT "SUBTOTALS",G$(4):IF G$(4)="
N"THEN 370:INPUT "BY ITEMS",A$
360GOSUB '16:IF I6=2THEN 220:T4=(D0-K9)/2:FOR I=1TO T4:G(I)=VAL(
P$(K9+2*I)):NEXT I:D0=K9
370INPUT "SUMMARY REPORT ONLY",G$(5)
380DATA SAVE OPEN "REPORT":DATA SAVE G$(),G(),T4
390T3=0
400IF C1$="N"THEN 410:PRINT T3+1;:INPUT "WIDTH;CONTENTS",A$:DATA
SAVE A$:GOTO 420
410DATA LOAD A$
420IF A$=" "THEN 430:T3=T3+1:L=NUM(A$):BIN(X3$(T3))=D0+1:CONVERT
STR(A$,1,L)TO X8:BIN(X2$(T3))=X8:A$=STR(A$,L+2):GOSUB '16:IF I6=
2THEN 220:GOTO 400
430BIN(X3$(T3+1))=D0+1:IF G$(1)="N"THEN 450:IF C1$="N"THEN 510:P
RINT "COL HEADING":FOR I=1TO T3:S$(I,1),S$(I,2),S$(I,3)=" ":PRIN
T I;:INPUT A$:K=0
433K=K+1:L=POS(A$="/"):IF L=0THEN 436:S$(I,K)=STR(A$,1,L-1):A$=S
TR(A$,L+1):GOTO 433
436S$(I,K)=A$
438NEXT I
450PRINT "COLUMNS-";:IF G$(3)="N"THEN 460:PRINT "TOTALS;";
460PRINT " NO OF DECIMAL PLACES":FOR I=1TO T3:STR(H$(I),1)="CN":
BIN(STR(H$(I),3))=0:IF VAL(P$(VAL(X3$(I))))>2THEN 500:STR(H$(I),
1,1)="I"
470PRINT I;:INPUT A$:IF G$(3)="N"THEN 480:IF STR(A$,1,1)="N"THEN
490:STR(H$(I),2,1)="Y":GOTO 490
480IF A$=" "THEN 500:GOTO 495
490L=POS(A$=";"):IF L=0THEN 500:A$=STR(A$,L+1)
495CONVERT A$TO X8:BIN(STR(H$(I),3))=X8:STR(H$(I),1,1)="N"
500NEXT I:DATA SAVE S$(),H$():GOTO 520
510DATA LOAD S$(),H$()
520D0=K9:REWIND :IF H$<>"T"THEN 525:STOP "SELECT PRINTER AND TYP
E CONTINUE":GOTO 530
525STOP "LOAD DATA TAPE AND TYPE CONTINUE":REWIND :DATA SAVE OPE
N "STRUCT":DATA SAVE H$,I9,F1$,F7,F$(),F0 DATA SAVE END :DATA SA
VE OPEN "FILE":C1=0:I9=0:X3=0:FOR I=1TO T3:X3=X3+VAL(X2$(I)):NEX
T I
527F7=INT((X3-1)*.05)+1:T4=0:G$(5)="Y":G$(3),G$(1)="N":GOTO 560
```

Figure B-5. Listing of RETR4.

```
530REM *INIT:T1=0:M$(1)="^^^":FOR J=1TO 4:FOR I=1TO 5:X4(I,J)=0:
NEXT I:NEXT J:IF T4=0THEN 540:X5$(1)="  *":X5$(2)="  **":X5$(3)="
***"
540IF G$(1)<>"Y"THEN 560:GOSUB '91
560FOR S1=1TO R6:GOSUB '5(R0(S1)):FOR S2=R0(S1)TO R1(S1):GOSUB
7:C=0:IF I4=0THEN 610:I9=I9+F7
570IF T4=0THEN 600:IF M$(1)<>"^^^"THEN 580
575FOR J=1TO T4:K=G(J):M$(J)=B$(K):GOSUB '92(J,K):NEXT J:PRINT
T2=T2+1:GOTO 600
580L1=T4+1:FOR L0=T4TO 1STEP -1:K=G(L0):IF M$(L0)=B$(K)THEN 590:
L1=L1-1:PRINT :PRINT X5$(L0);"TOTAL FOR ";STR(F$(K),1,8);"  ";M$
(L0):T2=T2+2:M$(L0)=B$(K):GOSUB '96
590NEXT L0:PRINT :GOSUB '95:IF L1>T4THEN 600:FOR L=L1TO T4:GOSUB
'92(L,G(L)):NEXT L:PRINT :GOSUB '95
600S=20:GOSUB '97
610NEXT S2:NEXT S1:IF G$(3)<>"Y"THEN 615:PRINT "** GRAND TOTAL *
*":L=4:GOSUB '96
612SELECT PRINT 005(64):GOTO 100
615IF H$="T"THEN 612:DATA SAVE C1$():DATA SAVE END :REWIND :DATA
LOAD "STRUCT":FOR I=1TO T3:B$(I)=F$(I):STR(F$(I),9,1)=X2$(I):STR
(F$(I),10,1)=H$(I):STR(F$(I),11)=STR(H$(I),3)
617STR(F$(I),1,8)=S$(I,1):NEXT I:DATA RESAVE H$,I9,F1$,F7,F$(),T
3:REWIND :FOR I=1TO T3:F$(I)=B$(I):NEXT I:GOTO 100
620DEFFN'91:T1=T1+1:PRINT HEX(0C):FOR I=1TO 6:PRINT :NEXT I:PRIN
T "PAGE";T1:PRINT "DATE: ";F1$:PRINT "DATABASE: ";F$:PRINT "REPO
RT FORM: ";E$:PRINT :T2=11
630IF G$(1)="N"THEN 640:FOR J=1TO 3:L=0:FOR I=1TO T3:L=L+VAL(X2$
(I))+1:PRINT S$(I,J);TAB(L);:NEXT I:PRINT :NEXT J:PRINT
640RETURN
650DEFFN'92(J,K):PRINT X5$(J);" ";STR(F$(K),1,8);"  ";B$(K):T2=T
2+1:RETURN
660DEFFN'93(K,Y):N=VAL(STR(H$(K),3)):GOSUB '29(Y):GOSUB '94(K):F
ETURN
710DEFFN'94(T):C$="  ":STR(C$,1+VAL(X2$(T))-LEN(B$))=B$:B$=C$:RET
URN
720DEFFN'95:IF G$(1)<>"Y"THEN 730:T2=T2+1:IF T2<56THEN 730:GOSUB
'91
730RETURN
740DEFFN'96:Q,P=0:FOR Z=1TO T3:IF STR(H$(Z),2,1)<>"Y"THEN 750:P=
P+1:GOSUB '93(Z,X4(P,L)):FOR I=LTO T4:X4(P,I)=0:NEXT I:PRINT TAB
(Q);B$;
750Q=Q+VAL(X2$(Z))+1:NEXT Z:PRINT :GOSUB '95:RETURN
760DEFFN'97:P,W=0:FOR I=1TO T3:GOSUB '19(VAL(X3$(I)),VAL(X3$(I+1
))-1):IF STR(H$(I),1,1)<>"C"THEN 770:B$=D$(E2):E2=E4:GOTO 780
770GOSUB '93(I,E(E1)):IF STR(H$(I),2,1)<>"Y"THEN 780:P=P+1:CONVE
```

Figure B-5. Listing of RETR4 (Cont'd).

55

```
RI B$TO X8:X4(P,4)=X4(P,4)+X8:IF T4=0THEN 780:FOR L=1TO T4:X4(P,
L)=X4(P,L)+X8:NEXT L
780IF H$="T"THEN 788:M=VAL(X2$(I)):X9=1
781S=S+1:IF S<21THEN 783:C1=C1+1:S=1:IF C1<13THEN 783:DATA SAVE
C1$():C1=1
783I8=M:IF S+M-1<21THEN 784:I8=21-S
784M=M-I8
785STR(C1$(C1),S,I8)=STR(B$,X9,I8):X9=X9+I8:S=S+I8-1:IF M>0THEN
781
788IF G$(5)="Y"THEN 790:PRINT TAB(W);B$;:W=W+VAL(X2$(I))+1
790NEXT I:IF G$(5)="Y"THEN 805:PRINT :IF G$(2)<>"Y"THEN 800:PRIN
T :T2=T2+1
800GOSUB 795:RETURN
805RETURN
```

Figure B-5. Listing of RETR4 (Cont'd).

56

DISTRIBUTION

DEFENSE DOCUMENTATION CENTER
CAMERON STATION, BUILDING 5
ALEXANDRIA, VA 22314
 ATTN DDC-TCA (12 COPIES)

COMMANDER
USA RSCH & STD GP (EUR)
BOX 65
FPO NEW YORK 09510
 ATTN LTC JAMES M. KENNEDY, JR.
      CHIEF, PHYSICS & MATH BRANCH

COMMANDER
US ARMY MATERIEL DEVELOPMENT
 & READINESS COMMAND
5001 EISENHOWER AVENUE
ALEXANDRIA, VA 22333
 ATTN DRXAM-TL, HQ TECH LIBRARY

COMMANDER
USA ARMAMENT COMMAND
ROCK ISLAND, IL 61201
 ATTN DRSAR-ASF, FUZE DIV
 ATTN DRSAR-RDF, SYS DEV DIV - FUZES

COMMANDER
USA MISSILE & MUNITIONS
 CENTER & SCHOOL
REDSTONE ARSENAL, AL 35809
 ATTN ATSK-CTD-F

DIRECTOR
DEFENSE NUCLEAR AGENCY
WASHINGTON, DC 20305
 ATTN APTL, DASA TECH LIBRARY

DIRECTOR OF DEFENSE RES
 AND ENGINEERING
WASHINGTON, DC 20301
 ATTN TECHNICAL LIBRARY

DIRECTOR
NATIONAL SECURITY AGENCY
FORT GEORGE G. MEADE, MD 20755
 ATTN T. A. PRUGH

COMMANDER
US ARMY RESEARCH OFFICE (DURHAM)
P.O. BOX 12211
RESEARCH TRIANGLE PARK, NC 27709
 ATTN CRD-AA-IP

COMMANDER
USA ELECTRONICS COMMAND
FORT MONMOUTH, NJ 07703
 ATTN DRSEL-CE, COMMUNICATIONS-
      ELECTRONICS INTEGRATION OFC
 ATTN DRSEL-TL, ELECTRONICS TECHNOLOGY
      & DEVICES LABORATORY
 ATTN DRSEL-WL, ELECTRONIC WARFARE LAB
 ATTN DRSEL-GG, COMPUTER-AIDED DESIGN
      & ENGINEERING OFFICE
 ATTN DRSEL-GG, TECHNICAL LIBRARY

MOUNTAIN VIEW OFFICE (DRSEL-WL-RU)
ELECTRONIC WARFARE LABORATORY
P.O. BOX 205
MOUNTAIN VIEW, CA 94040

COMMANDER
USA MISSILE COMMAND
REDSTONE ARSENAL, AL 35809
 ATTN DRSMI-RBLD, CHIEF DOC SECTION

COMMANDER
USA MOBILITY EQUIPMENT R&D CENTER
FORT BELVOIR, VA 22060
 ATTN SMEFB-W, TECHNICAL LIBRARY

COMMANDER
EDGEWOOD ARSENAL
EDGEWOOD ARSENAL, MD 21010
 ATTN SMUEA-TS-L, TECH LIBRARY

COMMANDER
FRANKFORD ARSENAL
BRIDGE & TACONY STREETS
PHILADELPHIA, PA 19137
 ATTN K1000, TECHNICAL LIBRARY

COMMANDER
PICATINNY ARSENAL
DOVER, NJ 07801
 ATTN SARPA-TS-T-S, TECHNICAL LIBRARY

COMMANDER
USA ABERDEEN PROVING GROUND
ABERDEEN PROVING GROUND, MD 21005
 ATTN STEAP-TL, TECH LIBRARY, BLDG 305

COMMANDER
USA ELECTRONICS PROVING GROUND
FORT HUACHUCA, AZ 85613
 ATTN STEEP-PA-I, TECH INFO CENTER

COMMANDER
YUMA PROVING GROUND
YUMA, AZ 85364
 ATTN STEYP-MTL, TEST ENGINEERING DIV

COMMANDER
USA WEAPONS COMMAND, HA
ROCK ISLAND, IL 61201
 ATTN SWERR-PL, TECHNICAL LIBRARY

CHIEF OF NAVAL OPERATIONS
NAVY DEPARTMENT
WASHINGTON, DC 20350
 ATTN NOP-098, DIR, OFC OF RES, DEV,
      TEST, AND EVALUATION
 ATTN NOP-985F, WEAPONS TECH BR

COMMANDER
NAVAL ELECTRONICS LABORATORY CENTER
SAN DIEGO, CA 92152
 ATTN TECHNICAL LIBRARY

COMMANDER
PACIFIC MISSILE RANGE
NAVAL MISSILE CENTER
POINT MUGU, CA 93042
  ATTN CODE 5632, TECHNICAL LIBRARY

COMMANDER
NAVAL SURFACE WEAPONS CENTER
WHITE OAK, MD 20910
  ATTN L-315, TECH LIBRARY

COMMANDER
NAVAL SEA SYSTEMS COMMAND
2521 JEFFERSON DAVIS HIGHWAY
ARLINGTON, VA 20360
  ATTN NSEA-0632, LIBRARY BRANCH

DIRECTOR
NAVAL RESEARCH LABORATORY
WASHINGTON, DC 20390
  ATTN 2620, TECHNICAL LIBRARY BR

COMMANDER
NAVAL SHIP SYSTEMS COMMAND, HQ
2531 JEFFERSON DAVIS HIGHWAY
WASHINGTON, DC 20360
  ATTN NSHP-2052, TECH LIBRARY BR

COMMANDER
NAVAL WEAPONS CENTER
CHINA LAKE, CA 93555
  ATTN CODE 753, LIBRARY DIV

COMMANDER
NAVAL SURFACE WEAPONS CENTER
DAHLGREN, VA 22448
  ATTN TECHNICAL LIBRARY

US AIR FORCE, HEADQUARTERS
DCS, RESEARCH & DEVELOPMENT
WASHINGTON, DC 20330

COMMANDER
HQ AIR FORCE SYSTEMS COMMAND
ANDREWS AFB
WASHINGTON, DC 20331
  ATTN DAPL, TECHNICAL LIBRARY
  ATTN DPSL, TECH LIBRARY

COMMANDER
AF CAMBRIDGE RESEARCH LABORATORIES, AFSC
L. G. HANSCOM FIELD
BEDFORD, MA 01730
  ATTN E. CZERLINSKY

COMMANDER
ARMAMENT DEVELOPMENT AND TEST CENTER
EGLIN AIR FORCE BASE, FL 32542
  ATTN ADTC(DLOSL), TECH LIBRARY

COMMANDER
AERONAUTICAL SYSTEMS DIVISION, AFSC
WRIGHT-PATTERSON AFB, OH 45433
  ATTN ASD/SD, DEPUTY FOR SYSTEMS
  ATTN TECHNICAL LIBRARY

COMMANDER
HQ SPACE AND MISSILE SYSTEMS ORGANIZATION
P. O. 96960 WORLDWAYS POSTAL CENTER
LOS ANGELES, CA 90009
  ATTN SN, DEP FOR SPACE COMM SYS
  ATTN SYT, COMPUTER TECHNOLOGY OFC

COMMANDER
AF SPECIAL WEAPONS CENTER, AFSC
KIRTLAND AFB, NM 87117
  ATTN SWTSX, SURVIVABILITY/
       VULNERABILITY BRANCH

HQ, SAAMA, SANEPA
KELLEY AFB, TX 78241
  ATTN DIR OF MATERIEL MANAGEMENT

US ENERGY RESEARCH & DEVELOPMENT
  ADMINISTRATION
WASHINGTON, DC 20545
  ATTN TECHNICAL LIBRARY

DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS
WASHINGTON, DC 20234
  ATTN LIBRARY

LIBRARY OF CONGRESS
SCIENCE & TECHNOLOGY DIVISION
WASHINGTON, DC 20540
  ATTN HEAD, LIB OPNS,

NASA AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035
  ATTN S. J. DE FRANCE, DIRECTOR

NASA GEORGE C. MARSHALL SPACE FLIGHT CTR
HUNTSVILLE, AL 35812
  ATTN M-G & C-NS

NASA GODDARD SPACE FLIGHT CENTER
GREENBELT, MD 20771
  ATTN LIBRARY

NASA LEWIS RESEARCH CENTER
21000 BROOKPARK ROAD
CLEVELAND, OH 44135
  ATTN LIBRARIAN

COMMANDER
ROME AIR DEVELOPMENT CENTER, AFSC
GRIFFISS AFB, NY 13440
  ATTN LTF, COMPUTER ENGINEERING BR
  ATTN TECHNICAL LIBRARY

NASA SCIENTIFIC & TECH INFO FACILITY
P. O. BOX 33
COLLEGE PARK, MD 20740
 ATTN ACQUISITIONS BR (S-AK/DL)

NATIONAL OCEANIC & ATMOSPHERIC ADM
ENVIRONMENTAL RESEARCH LABORATORIES
BOULDER, CO 80302
 ATTN LIBRARY, R-51, TECH REPORTS

CALIFORNIA INSTITUTE OF TECHNOLOGY
JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CA 91103
 ATTN TDS, LIBRARY MANAGER

UNIVERSITY OF CALIFORNIA
LAWRENCE RADIATION LABORATORY
BERKLEY, CA 94720
 ATTN LIBRARY, BUILDING 50, RM 134

UNIVERSITY OF CALIFORNIA
LOS ALAMOS SCIENTIFIC LABORATORY
P.O. BOX 1663
LOS ALAMOS, NM 87544
 ATTN R. GAWLER

UNIVERSITY OF FLORIDA
GAINSVILLE, FL 32603
 ATTN R. C. JOHNSON, JR.
 ATTN R. D. WALKER

UNIVERSITY OF ILLINOIS
DEPARTMENT OF MATHEMATICS
URBANA, IL 61801
 ATTN LAWRENCE A WHITE

UNIVERSITY OF MARYLAND
COMPUTER SCIENCE DEPARTMENT
COLLEGE PARK, MD 20741
 ATTN DR. YAOHAN CHU

UNIVERSITY OF MICHIGAN
INFRARED INFORMATION & ANALYSIS CENTER
ANN ARBOR, MI 48106
 ATTN WILLIAM L. WOLFE

BELL TELEPHONE LABORATORIES
WHIPPANY ROAD
WHIPPANY, NJ 07981
 ATTN LIBRARIAN

TYMSHARE INC.
1911 NORTH FORT MYER DRIVE
ARLINGTON, VA 22209
 ATTN CARLYLE REEDER
 ATTN TOBY REUT

WANG LABORATORIES
8360 NORTH STREET
TEWKSBURY, MA 01876
 ATTN JASON TAYLOR
 ATTN HAROLD KOPLOW
 ATTN ROY KOLK

ILLINOIS STATE WATER SURVEY
BOX 232
URBANA, IL 61801
 ATTN MARIE F. BURNS, LIBRARIAN

DIGITAL ACOUSTICS, INC.
1415 E. McFADDEN, SUITE F
SANTA ANA, CA 92705
 ATTN MISS PAMELA HURST

COMMANDER
AFATL/DLRD
EGLIN AFB, FL 32542
 ATTN MR. COLLINS

NATIONAL INSTITUTES OF HEALTH
BETHESDA, MD 20014
 ATTN DR. C. PATLAK, BLDG 13, RM 1D24

MGR SYSTEMS
1510 RICHARDS AVENUE
WILLIAMSPORT, PA 17701
 ATTN MR. BURNETT TYSON

CHIEF ENGR COM-PU-TOR
76 OX YOKE DRIVE
WETHERSFIELD, CT 06109
 ATTN WM H. SMYERS, JR.

THE JOHNS HOPKINS UNIVERSITY
DEPARTMENT OF CHEMISTRY
BALTIMORE, MD 21218
 ATTN DR. JOYCE J. KAUFMAN
 ATTN MR. HARRY J. T. PRESTON

NAVAL SECURITY ENGINEERING FACILITY
3801 NEBRASKA AVENUE
WASHINGTON, DC 20390
 ATTN MR. JOHN H. BICKFORD, CODE 0245

KINNEY SHOE CORPORATION
16TH FLOOR
233 BROADWAY
NEW YORK, NY 10007
 ATTN MR. GEORGE J. MICHELSON
       SPEC CONS TO THE PRES

ICHTHYOLOGICAL ASSOCIATES, INC.
SCHUYLKILL RIVER ECOLOGICAL STUDY
FRICKS LOCK ROAD, RD 1
POTTSTOWN, PA 19464
 ATTN MR. KEN LITE

BLECK ENGINERRING CO., INC.
1321 GLEN ROCK AVENUE
WAUKEGAN, IL 60085
 ATTN DONNA L. BLECK

DIRECTOR OF LABORATORIES
HOLY CROSS HOSPITAL
2701 WEST 68TH ST
 AT CALIFORNIA AVENUE
CHICAGO, IL 60629
 ATTN A. M. RING, MD

J. J. GARCIA & ASSOCIATES, INC.
11039 N. E., 6TH AVENUE
MIAMI, FL 33161
 ATTN ENRIQUE ALVAREZ

LINE COUPLING EQUIPMENT ENGINEERING
GENERAL ELECTRIC COMPANY
MOUNTAIN VIEW ROAD
LYNCHBURG, VA 24502
 ATTN MR. D. B. BRAH, MANAGER

OREGON STATE UNIVERSITY
SCHOOL OF OCEANOGRAPHY
CORVALLIS, OR 97331
 ATTN DR. LOUIS I. GORDON
       ASSISTANT PROFESSOR

CARTER PRODUCTS
RESEARCH LABORATORY
CRANBURY, NJ 08512
 ATTN W. M. WOODING
       DIRECTOR, TECHNICAL SERVICES

PICKARD & ANDERSON ENGINEERS
69 SOUTH ST
AUBURN, NJ 13021
 ATTN WILLIAM C. ANDERSON, P.E.

NORTHEASTERN PRODUCTS COMPANY
3500 S. CLINTON AVENUE
SOUTH PLAINFIELD, NJ 07080
 ATTN RICHARD D. GUIDO
       MANAGER QUALITY CONTROL

COMMANDANT
USA FIELD ARTILLARY SCHOOL
FORT SILL, OK 73503
 ATTN BILL MILLSPAUGH
       ATSFCTD/SD

CALCULATOR CONSULTANT
45-3A MT. PLEASANT VILLAGE
MORRIS PLAINS, NJ 07950
 ATTN MR. NEAL H. KUHN

US ENVIRONMENTAL PROTECTION AGENCY
P.O. BOX 5036
ROCHESTER, NY 14627
 ATTN MR. DONALD J. CASEY
       CHIEF, IFYGL BRANCH

ST. MARY'S UNIVERSITY
2700 CINCINNATI AVENUE
SAN ANTONIO, TX 28284
 ATTN DR. TOM MOTE

ETHYL CORPORATION
FUNDAMENTAL STUDIES DEPARTMENT
TERRE HAUTE, IN 47808
 ATTN MR. CHARLES FURLAND

BOEING AEROSPACE COMPANY
P.O. BOX 3999
SEATTLE, WA 98124
 ATTN MR. MALCOLM MATHEWS
       MS 8C-41

PENNWALT CORPORATION
TECHNOLOGICAL CENTER
900 FIRST AVENUE
KING OF PRUSSIA, PA 19406
 ATTN DR. J. E. DOHANY

MR. L. H. CHAMBERLIN
10510 SUNNYBROOK LANE, SW
TACOMA, WA 98498

WANG LABORATORIES
4824 BOILING BROOK PARKWAY
ROCKVILLE, MD 20852
 ATTN RAY DORRIS

HARRY DIAMOND LABORATORIES
 ATTN MCGREGOR, THOMAS, COL, COMMANDING
       OFFICER/FLYER, I.N./LANDIS, P.E./
       SOMMER, H./CONRAD, E.E.
 ATTN CARTER, W.W., DR., ACTING TECHNICAL
       DIRECTOR/MARCUS, S.M.
 ATTN KIMMEL, S., IO
 ATTN CHIEF, 0021
 ATTN CHIEF, 0022
 ATTN CHIEF, LAB 100
 ATTN CHIEF, LAB 200
 ATTN CHIEF, LAB 300
 ATTN CHIEF, LAB 400
 ATTN CHIEF, LAB 500
 ATTN CHIEF, LAB 600
 ATTN CHIEF, DIV 700
 ATTN CHIEF, DIV 800
 ATTN CHIEF, LAB 900
 ATTN CHIEF, LAB 1000
 ATTN RECORD COPY, BR 041
 ATTN HDL LIBRARY (3 COPIES)
 ATTN CHAIRMAN, EDITORIAL COMMITTEE
 ATTN CHIEF, 047
 ATTN TECH REPORTS, 013
 ATTN PATENT LAW BRANCH, 071
 ATTN MCLAUGHLIN, P.W., 741
 ATTN LANHAM, C., PROGRAM & PLANS
 ATTN CLASEN, S. M., 120
 ATTN CHIEF, 310
 ATTN MANION, F. M., 310
 ATTN DRZEWIECKI, T., 310
 ATTN SPYROUPOULOS, C., 310
 ATTN CHIEF, 340
 ATTN GOTO, J., 340
 ATTN MON, G., 340
 ATTN INGERSOLL, P., 430 (3 COPIES)
 ATTN OVERMAN, D. L., 420
 ATTN BUTLER, R., 0025
 ATTN HINE, R., 0025

DISTRIBUTION (Cont'd)

HARRY DIAMOND LABORATORIES (Cont'd)
 ATTN JOHNSON, P., 610
 ATTN RAVILIOUS, C., 310
 ATTN ROSEN, R., 800
 ATTN WICKLUND, J., 280
 ATTN MARROLETTI, J., 0025
 ATTN MATHEWS, H. J., 0025
 ATTN OFFICE, 0025 FILE
 ATTN BETTWY, D., 710
 ATTN FUNKE, M., 340
 ATTN LIEBERMAN, S., 350
 ATTN CROSBY, T., 330
 ATTN DAVIDSON, H., 410
 ATTN HOPP, T. H., 930
 ATTN ISEMAN, J., 930
 ATTN HUFMAN, A., 0025
 ATTN HAUSNER, A., 0025
 ATTN BLOOM, H. (30 COPIES)